

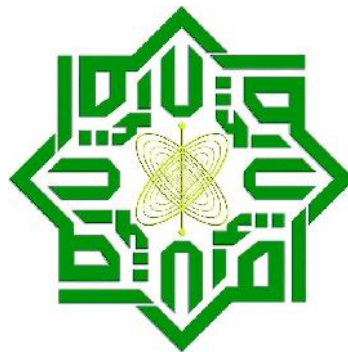
**APLIKASI LOCATION BASEDSERVICE (LBS) UNTUK  
PENCARIAN RUTE  
TERPENDEKMENGGUNAKANALGORITMA DIJKSTRA  
STUDI KASUS : PT. COCA COLA AMATIL INDONESIA  
SALES OFFICE PEKANBARU**

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Pada  
Jurusan Teknik Informatika

Oleh :

**LUTHFI FAHRONZI  
10751000378**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU  
PEKANBARU  
2013**

# **LEMBAR PERSETUJUAN**

## **APLIKASI *LOCATION BASED SERVICE* (LBS) UNTUK PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA**

### **STUDI KASUS : PT. COCA COLA AMATIL INDONESIA SALES OFFICE PEKANBARU**

#### **TUGAS AKHIR**

Oleh:

**LUTHEI FAHRONZI**  
**10751000378**

Telah diperiksa dan disetujui sebagai laporan tugas akhir  
di Pekanbaru, pada tanggal 03 April 2013

Koordinator Tugas Akhir

Pembimbing

**Reski Mai Candra, S.T., M.Sc**  
**NIK. 130 510 032**

**Novriyanto, S.T., M.Sc**  
**NIP. 19771128 200710 1 003**

## **LEMBAR PENGESAHAN**

### **APLIKASI *LOCATION BASED SERVICE* (LBS) UNTUK PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA**

#### **STUDI KASUS : PT. COCA COLA AMATIL INDONESIA SALES OFFICE PEKANBARU**

### **TUGAS AKHIR**

Oleh:

**LUTHFI FAHRONZI**  
**10751000378**

Telah dipertahankan di depan sidang dewan penguji  
Sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika  
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau  
di Pekanbaru, pada tanggal 03 April 2013

Pekanbaru, 03 April 2013  
Mengesahkan,

**Dekan**

**Ketua Jurusan**

**Dra. Hj. Yenita Morena, M.Si**  
**NIP. 19601125 198503 2 002**

**Novriyanto, S.T., M.Sc**  
**NIP. 19771128 200710 1 003**

#### **DEWAN PENGUJI**

Ketua : Novriyanto, S.T., M. Sc \_\_\_\_\_

Sekretaris : Novriyanto, S.T., M. Sc \_\_\_\_\_

Anggota I : Benny Sukma Negara, S.T., M.T \_\_\_\_\_

Anggota II : M. Irsyad, M.T \_\_\_\_\_

## **LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL**

Tugas Akhir yang tidak diterbitkan ini terdaftar dan tersedia di Perpustakaan Universitas Islam Negeri Sultan Syarif Kasim Riau adalah terbuka untuk umum dengan ketentuan bahwa hak cipta pada penulis. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau ringkasan hanya dapat dilakukan seizin penulis dan harus disertai dengan kebiasaan ilmiah untuk menyebutkan sumbernya.

Penggandaan atau penerbitan sebagian atau seluruh Tugas Akhir ini harus memperoleh izin dari Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau. Perpustakaan yang meminjamkan Tugas Akhir ini untuk anggotanya diharapkan untuk mengisi nama, tanda peminjaman dan tanggal pinjam.

## **LEMBAR PERNYATAAN**

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan didalam daftar pustaka.

Pekanbaru, 03 April 2013

Yang membuat pernyataan,

**LUTHFI FAHRONZI**

## LEMBAR PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٥﴾

*" Karena sesungguhnya sesudah kesulitan itu ada kemudahan "*

{QS. Alam Nasyrat (5)}

***Syukur Alhamdulillah penulis ucapkan kepada Allah swt. atas limpahan rahmat dan karunia-Nya.***

Tugas akhir ini penulis persembahkan untuk Bapak dan Mak tercinta, H. Abas dan Hj. Rosilawati Dewi yang selalu mendo'akan dan memberikan dukungannya yang tiada henti. *"Sungguh sangat besar kasih sayang dan pengorbanan yang kalian berikan untuk kesuksesan masa depan ku. Terima kasih Bapak dan Mak, semua yang telah kalian berikan tidak akan pernah terbalas, terlupakan, dan tidak akan pernah tergantikan".*

***Kemudian untuk keluarga besar penulis di Kecamatan Durai, untuk sahabat dan teman-teman penulis yang tidak dapat disebutkan satu persatu. Jerimakasih atas bantuan dan dukungannya yang berharga.***

# ***LOCATION BASED SERVICE (LBS) APPLICATION FOR SHORTEST ROUTE SEARCHING BY DIJKSTRA ALGORITHM***

***CASE STUDY : PT. COCA COLA AMATIL INDONESIA  
SALES OFFICE PEKANBARU***

**LUTHFI FAHRONZI**

**10751000378**

*Final Exam Date: April 03<sup>th</sup>, 2013*

*Graduation Ceremony Period: Juni 2013*

*Information Engineering Department*

*Faculty of Sciences and Technology*

*State Islamic University of Sultan Syarif Kasim Riau*

## ***ABSTRACT***

*Location based service (LBS) have ability for searched the informations about phisical locations, and have features for supporting the finding routes to some destinations. PT. Coca Cola Amatil Indonesia Sales Office Pekanbaru (PT. CCAI SOP) is a marketing and supplier company for packaging drinks products. Customer availability could be as the optional or alternative for specfied the objective customer routes. On this recent times the customer routes searched was done by sense, so the gained outputs not right yet. Therefore required an location based services (LBS) application for shortest route searching client server based on Android platform. This application was expected to provide informations about the structural customer sequence from the nearest one to the farthest one that should be visited using Dijkstra algorithm, and provide the route informatios of the streets that could be passed using Google Maps. The outputs from CCAI application using Dijkstra algorithm presents the customer list that targeted point from start to end destinations, and presents the street routes from start location to destination location. For more CCAI application development could be implemented on multy platform.*

***Keywords:****Android, Client, Dijkstra, Google Maps, Location Based Service (LBS), PT. Coca ColaAmatil Indonesia Sales Office Pekanbaru (PT. CCAI SOP), Server.*

# **APLIKASI *LOCATION BASED SERVICE* (LBS) UNTUK PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA**

## **STUDI KASUS : PT. COCA COLA AMATIL INDONESIA SALES OFFICE PEKANBARU**

**LUTHFI FAHRONZI**

**10751000378**

Tanggal Sidang: 03 April 2013

Periode Wisuda: Juni 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

### **ABSTRAK**

*Location based service* (LBS) atau layanan berbasis lokasi memiliki kemampuan untuk mencari informasi tentang lokasi fisik, dan memiliki fitur yang mendukung temuan rute ke tujuan tertentu. PT. Coca Cola Amatil Indonesia Sales Office Pekanbaru (PT. CCAI SOP) bergerak dibidang pemasaran dan penyedia dari produk-produk minuman dalam kemasan. Tersedianya banyak *customer* dapat dijadikan sebagai pilihan atau alternatif untuk menentukan rute *customer* yang akan dituju. Pada saat ini penelusuran rute *customer* dilakukan secara nalar, sehingga hasil yang diperoleh belum tepat. Untuk itu dibutuhkan suatu aplikasi *location based services* (LBS) untuk pencarian rute terpendek berbasis *client server* pada *platform* Android. Aplikasi ini diharapkan dapat memberikan informasi mengenai urutan *customer* yang terstruktur dari yang terdekat sampai yang terjauh yang harus dikunjungi menggunakan algoritma Dijkstra, dan memberikan informasi rute jalan yang bisa dilewati menggunakan *google maps*. Hasil pengujian aplikasi CCAI dengan menggunakan algoritma Dijkstra dapat menampilkan urutan daftar *customer* yang akan dituju dari awal hingga akhir, serta menampilkan rute jalan dari lokasi awal ke lokasi tujuan. Untuk pengembangan lebih lanjut aplikasi CCAI dapat digunakan pada multi *platform*.

**Kata kunci:** *Android, Client, Dijkstra, Google Maps, Location Based Service (LBS), PT. Coca Cola Amatil Indonesia Sales Office Pekanbaru (PT. CCAI SOP), Server.*



## KATA PENGANTAR

*Alhamdulillah Robbil'alamin*, puji syukur penulis ucapkanke-hadirat Allah SWT. Tuhan semesta alam, karena atas segala limpahan rahmat dan karuniah-Nyasehingga penulis dapat menyelesaikan penelitian tugas akhir ini. Selanjutnya shalawat serta salam penulis ucapkan kepada junjungan alam, Rasul Allah, yang menjadi suri tauladan bagi umatnya, yakni Nabi Muhammad SAW. dengan ucapan *Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad*.

Laporan tugas akhir ini merupakan salah satu prasyarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar kesarjanaan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA Riau). Selama menyelesaikan tugas akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan, dan petunjuk dari banyak pihak baik secara langsung maupun tidak langsung. Untuk itu dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Dra. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Novriyanto, ST, M.Sc, selaku Ketua Jurusan Teknik Informatika, sekaligus sebagai Dosen Pembimbing I Tugas Akhir. Terimakasih atas segala dukungan, nasehat, kesabaran, dan motivasi yang telah Bapak berikan.
4. Reski Mai Candra, ST, M.Sc, selaku Koordinator Tugas Akhir. Terima kasih atas bantuan Bapak dalam mempersiapkan semua kebutuhan penulis untuk penyelesaian Tugas Akhir.
5. Benny Sukma Negara, MT, selaku Dosen Penguji 1. Terimakasih juga untuk ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukan yang telah Bapak berikan untuk penyempurnaan laporan ini.

6. M. Irsyad, MT, selaku Dosen Penguji 2. Terimakasih atas ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukan yang telah Bapak berikan untuk penyempurnaan laporan ini.
7. Dan terakhir, terimakasih pula penulis ucapkan untuk Almamater Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu. Terimakasih banyak atas bantuan dan dukungannya yang berharga.

Akhirnya, penulis menyadari dalam penulisan laporan ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik sangat penulis harapkan untuk kemajuan penulis secara pribadi. Terimakasih.

Pekanbaru, 03 April 2013

Penulis

## DAFTAR ISI

HALAMAN JUDUL LAPORAN .....	i
LEMBAR PERSETUJUAN .....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN .....	v
LEMBAR PERSEMBAHAN .....	vi
ABSTRAK .....	vii
<i>ABSTRACT</i> .....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL .....	xviii
DAFTAR LAMPIRAN .....	xix
DAFTAR SIMBOL .....	xxi
DAFTAR ISTILAH .....	xxii

### BAB I PENDAHULUAN

1.1. Latar Belakang .....	I-1
1.2. Rumusan Masalah .....	I-3
1.3. Batasan Masalah .....	I-3
1.4. Tujuan Penelitian .....	I-3
1.5. Manfaat Penelitian .....	I-3
1.6. Sistematika Penulisan .....	I-4

### BAB II LANDASAN TEORI

2.1. <i>Location-Based Services</i> (LBS) .....	II-1
---	------

2.1.1. Definisi LBS .....	II-1
a. <i>Push Services</i> .....	II-1
b. <i>Pull Services</i> .....	II-2
2.1.2. <i>LBS Components</i> .....	II-3
a. <i>Mobile Devices</i> .....	II-3
b. <i>Communication Network</i> .....	II-3
c. <i>Positioning Component</i> .....	II-3
d. <i>Service and Application Provider</i> .....	II-3
e. <i>Data and Content Provider</i> .....	II-4
2.1.3. <i>Types of Location-Based Services</i> .....	II-5
a. <i>Emergency Services</i> .....	II-6
b. <i>Navigation Services</i> .....	II-6
c. <i>Information Services</i> .....	II-6
d. <i>Advertising Services</i> .....	II-6
e. <i>Tracking Services</i> .....	II-6
f. <i>Billing Services</i> .....	II-6
2.2. Algoritma Dijkstra .....	II-7
2.2.1. Definisi Algoritma Dijkstra .....	II-7
2.2.2. Pengenalan Algoritma Dijkstra .....	II-7
2.3. Google Maps .....	II-11
2.4. <i>Object Oriented Analysis and Design (OOAD)</i> .....	II-12
2.4.1. <i>Unified Modelling Language (UML)</i> .....	II-12
2.4.1.1. <i>Use Case Diagram</i> .....	II-13
2.4.1.2. <i>Class Diagram</i> .....	II-13
2.4.1.3. <i>Activity Diagram</i> .....	II-15
2.4.1.4. <i>Statechart Diagram</i> .....	II-15
2.4.1.5. <i>Sequence Diagram</i> .....	II-15

2.4.1.6. <i>Deployment Diagram</i> .....	II-16
2.5. <i>Rational Unified Process (RUP)</i> .....	II-16
2.5.1. Definisi RUP .....	II-16
2.5.2. Fase RUP .....	II-18
1. Fase <i>Inception</i> .....	II-18
2. Fase <i>Elaboration</i> .....	II-19
3. Fase <i>Construction</i> .....	II-20
4. Fase <i>Transition</i> .....	II-20

### **BAB III METODOLOGI PENELITIAN**

3.1. Tahapan Penelitian .....	III-1
3.2. Tahapan <i>Rational Unified Process (RUP)</i> .....	III-2
3.2.1. Fase <i>Inception</i> .....	III-2
3.2.2. Fase <i>Elaboration</i> .....	III-3
3.2.3. Fase <i>Construction</i> .....	III-3
3.2.4. Fase <i>Transition</i> .....	III-3

### **BAB IV ANALISA DAN PERANCANGAN**

4.1. Fase <i>Inception</i> .....	IV-1
4.1.1. Analisa Permasalahan .....	IV-1
4.1.2. Gambaran Umum Aplikasi .....	IV-2
4.1.3. Deskripsi Kebutuhan Sistem .....	IV-4
4.1.3.1. Sistem Yang Akan Dibangun .....	IV-4
a. Aplikasi Pada Perangkat Android .....	IV-5
b. <i>Connector</i> .....	IV-5
4.1.3.2. Analisa Fungsional .....	IV-6
4.1.3.3. Performansi Aplikasi .....	IV-6
4.1.4. Fungsi Sistem .....	IV-7
4.1.4.1. Fungsi Aplikasi dari Sisi Perangkat Android .....	IV-7

4.1.4.2. Fungsi <i>Database Server</i> .....	IV-7
4.1.5. Modul <i>Assisted GPS</i> , Modul Google Maps API, dan Modul JSON <i>Parsing</i> .....	IV-7
4.1.5.1. Modul <i>Assisted GPS</i> .....	IV-7
4.1.5.2. Modul Google Maps API.....	IV-8
4.1.5.3. Modul JSON <i>Parsing</i> .....	IV-8
4.1.6. Model Sistem .....	IV-8
4.1.7. Deskripsi Pengguna .....	IV-9
4.1.8. Perhitungan Manual Algoritma Dijkstra .....	IV-10
4.1.9. Perancangan Sistem .....	IV-18
4.1.9.1. Pemodelan UML ( <i>Unified Modeling Language</i> ) .....	IV-18
4.1.9.1.1. <i>Usecase Diagram</i> .....	IV-18
4.1.9.1.2. <i>Use Case Specification</i> .....	IV-19
4.1.9.1.3. <i>Class Diagram</i> .....	IV-20
4.1.9.1.4. <i>Activity Diagram</i> .....	IV-22
4.1.9.1.5. <i>Sequence Diagram</i> .....	IV-22
4.2. Fase <i>Elaboration</i> .....	IV-23
4.2.1. Perancangan Tabel <i>Database</i> .....	IV-23
4.2.1.1. Perancangan Tabel <i>User</i> (Pengguna).....	IV-23
4.2.1.2. Perancangan Tabel <i>Customer</i> (Pelanggan) .....	IV-24
4.2.2. Perancangan Struktur Menu Aplikasi.....	IV-25
4.2.3. Perancangan <i>Interface</i> Aplikasi.....	IV-25
4.2.3.1. Perancangan <i>Interface</i> Beranda (Daftar <i>Customer</i> ) ..	IV-25

## **BAB V IMPLEMENTASI DAN PENGUJIAN**

5.1. Fase <i>Construction</i> .....	V-1
5.1.1. Implementasi Aplikasi .....	V-1
5.1.1.1. Lingkungan Pengembangan .....	V-1

5.1.1.2. Lingkungan Implementasi .....	V-2
5.1.1.3. Implementasi <i>Interface</i> Aplikasi .....	V-3
5.2. Fase <i>Transition</i> .....	V-3
5.2.1. Pengujian Aplikasi .....	V-3
5.2.1.1. Pengujian <i>Black Box</i> .....	V-4
5.2.1.2. <i>User Acceptance Test</i> .....	V-13
5.2.2. Kesimpulan Pengujian .....	V-15

## **BAB VI PENUTUP**

6.1 Kesimpulan .....	VI-1
6.2 Saran .....	VI-2

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## **DAFTAR RIWAYAT HIDUP**

## DAFTAR GAMBAR

Gambar	Halaman
2.1. LBS sebagai Persimpangan Teknologi .....	II-2
2.2. Komponen dasar LBS .....	II-4
2.3. Komponen dan Proses LBS .....	II-5
2.4. Graf Berarah .....	II-10
2.5. Hasil Penerapan Algoritma Dijkstra .....	II-11
2.6. Dua Dimensi RUP .....	II-17
3.1. Tahapan Penelitian dengan Metode RUP .....	III-1
4.1. Arsitektur Sistem .....	IV-3
4.2. Model Sistem .....	IV-8
4.3. Graf Berbobot .....	IV-10
4.4. Langkah 1 .....	IV-13
4.5. Langkah 2 .....	IV-13
4.6. Langkah 3 .....	IV-14
4.7. Langkah 4 .....	IV-14
4.8. Langkah 5 .....	IV-15
4.9. Langkah 6 .....	IV-15
4.10. Langkah 7 .....	IV-16
4.11. Langkah 8 .....	IV-16
4.12. Langkah 9 .....	IV-17
4.13. Langkah 10 .....	IV-17
4.14. Rute Terpendek .....	IV-18
4.15. Aliran <i>Usecase Diagram</i> .....	IV-19
4.16. <i>Class Diagram</i> .....	IV-21
4.17. <i>Activity Diagram Login</i> .....	IV-22
4.18. <i>Sequence Diagram</i> Menampilkan Rute ke Lokasi <i>Customer</i> .....	IV-23
4.19. Rancangan Struktur Menu Aplikasi .....	IV-25
4.20. Perancangan <i>Interface</i> Beranda (Daftar <i>Customer</i> ) .....	IV-26
5.1. Hasil Implementasi Halaman <i>Login</i> .....	V-3



5.2. Grafik Pengujian <i>User Acceptance Test</i> .....	V-15
B.1. <i>Activity Diagram Login</i> .....	B-1
B.2. <i>Activity Diagram Detail Customer</i> .....	B-2
B.3. <i>Activity Diagram Map</i> .....	B-3
B.4. <i>Activity Diagram About</i> .....	B-3
C.1. Perancangan <i>Interface Beranda (Daftar Customer)</i> .....	C-1
C.2. Perancangan <i>Interface Detail Customer</i> .....	C-2
C.3. Perancangan <i>Interface Map</i> .....	C-2
C.4. Perancangan <i>Interface Direction</i> .....	C-3
C.5. Perancangan <i>Interface About</i> .....	C-3
D.1. Hasil Implementasi Halaman <i>Login</i> .....	D-1
D.2. Hasil Implementasi Halaman Beranda (Daftar <i>Customer</i> ) .....	D-2
D.3. Hasil Implementasi Halaman <i>Detail Customer</i> .....	D-2
D.4. Hasil Implementasi Halaman <i>View Maps</i> .....	D-3
D.5. Hasil Implementasi Halaman Map .....	D-3
D.6. Hasil Implementasi Halaman <i>About</i> .....	D-4

## DAFTAR LAMPIRAN

Lampiran	Halaman
A. <u>Deskripsi Rinci</u> <i>Use Case Specification</i> .....	A-1
B. <u>Deskripsi Rinci</u> <i>Activity Diagram</i> .....	B-1
C. <u>Deskripsi Rinci</u> Rancangan <i>Interface</i> Aplikasi .....	C-1
D. Hasil Implementasi .....	D-1
E. Kuisioner Pengujian <i>User Acceptance Test</i> Aplikasi CCAI .....	E-1

## DAFTAR TABEL

Tabel	Halaman
4.1. Deskripsi Pengguna .....	IV-9
4.2. Jarak antar PT. CCAI dan <i>Customer</i> (Km) .....	IV-11
4.3. <i>Use Case Specification</i> Melihat Daftar <i>Customer</i> .....	IV-20
4.4. Tabel <i>User</i> (Pengguna) .....	IV-24
4.5. Tabel <i>Customer</i> (Pelanggan) .....	IV-24
5.1. Pengujian Fitur dan Fungsi Aplikasi CCAI .....	V-5
5.2. Pengujian Aplikasi dengan beberapa <i>Provider</i> Jaringan dan Pada beberapa Merk Perangkat Android .....	V-8
5.3. Pengujian Aplikasi untuk Menampilkan Rute Jalan ke Lokasi <i>Customer</i> .....	V-9
5.4. Kuisisioner <i>User</i> .....	V-13
5.5. Hasil Pengujian <i>User Acceptance Test</i> .....	V-14
A.1 <i>Usecase Specification</i> Menampilkan Map .....	A-1
A.2. <i>Usecase Specification Login</i> .....	A-2
A.3. <i>Usecase Specification</i> Menampilkan Detail <i>Customer</i> .....	A-2
A.4. <i>Usecase Specification</i> Menampilkan Informasi Koordinat .....	A-3
A.5. <i>Usecase Specification</i> Menampilkan Map Rute Jalan .....	A-4

## DAFTAR SIMBOL



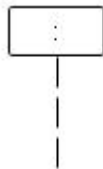
Aktor (Pengguna)



*Use Case* (Proses)



*Unidirectional Association*



*Object* (Objek)



*Class* (Kelas)



*Start State*(Status mulai)



*End State*(Status selesai)



*Activity*(Aktivitas)



*Decision*(Keputusan)



*Partision* (Partisi)



*Comment* (Pesan)

## DAFTAR ISTILAH

<i>Android</i>	: <i>Operating System</i> (OS) yang digunakan pada perangkat <i>smartphone</i>
<i>Client</i>	: Pengguna aplikasi
<i>Database</i>	: Basis data yang berfungsi sebagai media penyimpan data
<i>Dijkstra</i>	: Algoritma Pencarian Rute Terpendek
<i>Interface</i>	: Antarmuka
<i>Location Based Services (LBS)</i>	: Suatu layanan yang berbasis lokasi
<i>Server</i>	: Penyediaan layanan sistem
<i>Smartphone</i>	: Perangkat komunikasi yang memiliki sistem operasi

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Teknologi terbaru dari layanan *mobile* adalah *Location Based Service* (LBS) atau layanan berbasis lokasi. Teknologi nirkabel mempunyai peran yang sangat penting untuk kemajuan layanan ini. LBS memiliki kemampuan untuk mencari informasi tentang lokasi fisik, dan memiliki fitur yang mendukung temuan rute ke tujuan tertentu (Beaubrun *et al.*, 2007). Beberapa tahun terakhir LBS menjadi sumber baru bagi kepentingan operator *mobile* untuk meningkatkan penawaran layanan mereka, sehingga berpotensi meningkatkan pendapatan ([www.northstream.se](http://www.northstream.se), 2001).

PT. Coca Cola Amatil Indonesia Sales Office Pekanbaru (PT. CCAI SOP) bergerak dibidang pemasaran dan penyedia dari produk-produk minuman dalam kemasan seperti Coca Cola, Sprite, Fanta, Ades, dan Fres Tea. PT. CCAI melayani pemesanan untuk wilayah Pekanbaru, Kerinci, dan Ujung Batu. Untuk jumlah *customer* dari ketiga wilayah tersebut ada 2600 *customer*, jumlah tersebut sudah termasuk *customer* yang aktif dan non aktif. Jenis dari *customer* tersebut berbeda-beda, seperti pasar tradisional, restoran, *cafe*, *mall*, dan minimarket. Setiap 60 *customer* memiliki satu *Sales Representative* (SR) yang bertugas mengelola dan melayani pemesanan. PT. CCAI menggunakan 15 buah mobil khusus sebagai media pengantar pesanan *customer*. Pengantaran pesanan dilakukan setiap hari kerja dan terjadwal. Satu mobil menangani 10 *customer*, kondisi ini bisa berubah berdasarkan daftar pengantaran. Yang bertugas mengatur lokasi tujuan dari suatu mobil adalah *Supervisor Delivery Man*. Tersedianya banyak *customer* dapat dijadikan sebagai pilihan atau alternatif untuk menentukan rute *customer* yang akan dituju. Sistem kerja pada PT. CCAI yaitu dengan menggunakan pengetahuan karyawan sebagai pengendara mobil mengenai kota Pekanbaru dan alamat lokasi *customer* sebagai petunjuk, selanjutnya ditentukan urutan *customer* yang akan dikunjungi dari awal hingga akhir. Dari sistem kerja tersebut ditemukan suatu permasalahan, yaitu urutan *customer* yang diperoleh tidak mengacu pada suatu perhitungan yang akurat untuk masing-masing jarak antar *customer*, sehingga waktu yang diperoleh tidak efisien dan belum optimal.

Dari uraian diatas penulis tertarik melakukan penelitian tugas akhir mengenai aplikasi *location based services* (LBS) untuk pencarian rute terpendek dan PT. CCAI sebagai tempat studi kasus. Aplikasi ini diharapkan dapat menjadi salah satu solusi untuk menyelesaikan permasalahan dan memberikan informasi

mengenai urutan *customer* yang terstruktur dari yang terdekat sampai yang terjauh yang harus dikunjungi, dan memberikan informasi rute jalan yang bisa dilewati.

Penelitian serupa telah dilakukan oleh Dewi (2010), tentang pencarian rute terpendek tempat wisata di Bali dengan menggunakan algoritma Dijkstra. Hasilnya adalah algoritma Dijkstra memberikan solusi yang cukup baik untuk digunakan pada pencarian rute terpendek dari lokasi asal menuju ke lokasi wisata di Bali.

Penelitian juga telah dilakukan oleh Faizah Ifatul (2010), tentang rancang bangun perangkat lunak penentuan rute perjalanan wisata di Malang menggunakan algoritma Dijkstra. Hasil yang didapat adalah rekomendasi rute yang optimum dari lokasi asal ke lokasi tujuan.

Dalam penelitian ini, diusulkan suatu aplikasi *Location Based Service* (LBS) untuk pencarian rute terpendek menggunakan algoritma rute terpendek. Ada beberapa algoritma yang dapat digunakan dalam pencarian rute terpendek, seperti algoritma Dijkstra, A\* (A-Star), Bellman Ford, dan Floyd Warshall. Pada penelitian ini penulis akan menggunakan algoritma Dijkstra sebagai metode pencarian rute terpendek. Alasan memilih algoritma Dijkstra karena algoritma tersebut merupakan algoritma penelusuran graf yang menyelesaikan permasalahan rute terpendek dengan satu sumber asal untuk suatu graf dengan nilai sisi non negatif, menghasilkan pohon jalur terpendek. Algoritma ini sering digunakan dalam *routing* dan protokol jaringan lainnya yang terkait (Rutter, 2009).

## **1.2. Rumusan Masalah**

Berdasarkan dari latar belakang yang telah dijelaskan, maka penelitian ini akan membahas mengenai “Bagaimana membangun aplikasi *Location Based Service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra”.

## **1.3. Batasan Masalah**

Batasan masalah pada penelitian ini adalah:

1. Aplikasi *client* dijalankan pada *platform* Android.
2. Wilayah yang akan digunakan adalah kota Pekanbaru.
3. Tugas akhir ini tidak membahas aspek-aspek keamanan komunikasi data dari sisi *client* maupun *server*.
4. *Map* yang digunakan bersumber dari Google Maps.
5. Rute jalan dari lokasi awal ke lokasi tujuan diperoleh dari Google Maps



6. Waktu tempuh diperoleh dari asumsi kecepatan umum 40 km/ jam, tanpa mempertimbangkan kondisi jalan dan kemacetan.

#### **1.4. Tujuan Penelitian**

Tujuan yang ingin dicapai penulis dari penelitian Tugas Akhir ini adalah dibangun aplikasi *Location Based Service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra.

#### **1.5. Manfaat Penelitian**

Manfaat penelitian dan pembuatan aplikasi *Location Based Service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra adalah:

1. Memberikan informasi dalam menentukan urutan *customer* yang akan dikunjungi.
2. Meningkatkan pemahaman tentang struktur dan sistem kerja dalam membangun aplikasi *Location Based Service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra.

#### **1.6. Sistematika Penulisan**

Berikut merupakan rencana susunan sistematika penulisan laporan Tugas Akhir yang akan dibuat :

Bab I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan dari Tugas Akhir yang dibuat.

Bab II Landasan Teori

Bab ini membahas tentang landasan teori yang digunakan sebagai pendukung penelitian tugas akhir ini, diantaranya *Location Based Services* (LBS), Algoritma Dijkstra, Google Maps, *Object Oriented Analysis and Design* (OOAD), dan *Rational Unified Process* (RUP).

Bab III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian dengan metode *Rational Unified Process* (RUP), diantaranya fase *inception*, fase *elaboration*, fase *constuction*, fase *transition*.

#### Bab IV Analisa dan Perancangan

Bab ini berisi pembahasan mengenai analisa sistem dan perancangan sistem yang akan dibangun berdasarkan tahapan pada fase *inception* dan fase *elaboration*

#### Bab V Implementasi dan Pengujian

Bab ini berisi penjelasan mengenai implementasi dan pengujian aplikasi berdasarkan tahapan pada fase *construction* dan fase *transition*.

#### Bab VI Kesimpulan dan Saran

Bab ini berisi penjelasan tentang kesimpulan dan saran yang diambil dari hasil penelitian tugas akhir ini.

## BAB II

### LANDASAN TEORI

#### 2.1 *Location-Based Services (LBS)*

*Location based services (LBS)* atau layanan berbasis lokasi merupakan suatu layanan yang diberikan kepada pengguna layanan berdasarkan lokasi pengguna.

##### 2.1.1 Definisi LBS

LBS didefinisikan sebagai suatu layanan informasi dan hiburan yang dapat diakses dari perangkat *mobile* melalui jaringan *mobile*, dengan memanfaatkan lokasi terminal *mobile* (Azfar *et al.*, 2010). Definisi serupa disampaikan oleh Afwani dan Fiarni (2011) *Location Based Services (LBS)* sebagai layanan informasi dengan memanfaatkan teknologi untuk mengetahui suatu posisi. LBS ini menggunakan teknologi *Positioning System*, teknologi ini memungkinkan para pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya. LBS mengacu pada suatu aplikasi yang mengeksploitasi pengetahuan tentang posisi geografis dari perangkat *mobile* untuk memberikan layanan informasi (IAMAI, 2008).

LBS dapat diklasifikasikan berdasarkan jenisnya, seperti *push* dan *pull services* (Azfar *et al.*, 2010).

##### a. *Push Services*

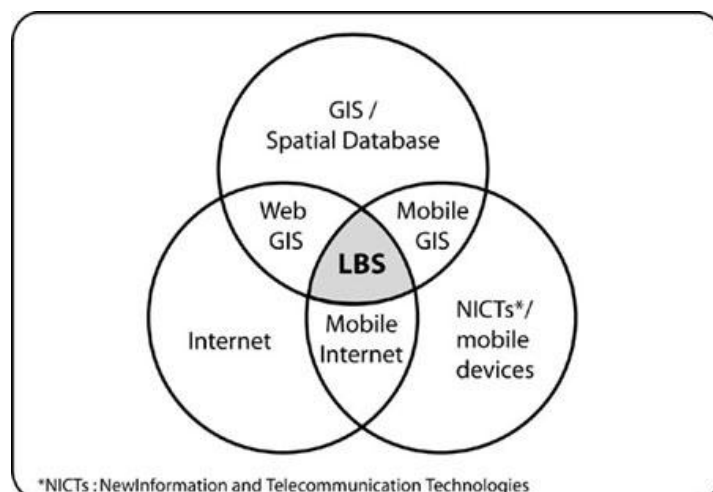
Pada *push services* pengguna layanan menerima informasi tanpa harus memintanya, dengan kata lain informasi tersebut secara proaktif dikirim ke pengguna layanan.

##### b. *Pull Services*

Pada *pull services* pengguna layanan aktif melakukan permintaan informasi melalui jaringan. Dalam *pull services* berbasis lokasi, pengguna melakukan pemeriksaan informasi secara berkala.

LBS termasuk dalam kategori teknologi yang sama dengan *geographic information system* (GIS), dan aplikasi *global positioning system* (GPS), yaitu dikenal dengan teknologi geospasial. Teknologi ini terdiri dari perangkat untuk mengumpulkan, menyimpan, menganalisa dan mendistribusikan data yang sesuai dengan kebutuhan pengguna terhadap sistem koordinat bumi secara *realtime*. Identifikasi kordinat pengguna memungkinkan aplikasi LBS untuk menyediakan layanan bagi pengguna perangkat *mobile*. Layanan ini menjadi sangat penting bagi penggunanya karena mampu menghubungkan antara lokasi informasi geografis terhadap lokasi penggunaanya, hal ini sangat mendukung era mobilitas seperti pada masa sekarang (Afwani dan Fiarni, 2011).

Keberadaan aplikasi LBS merupakan hasil penggabungan dari tiga buah teknologi yaitu New Information and Communication Technologies (NICTS) seperti sistem telekomunikasi *mobile* dan perangkat genggam, dari Internet, dan dari Sistem Informasi Geografis (SIG) dengan spasial database (Shiode *et al.*, 2002).



Gambar 2.1. LBS sebagai Persimpangan Teknologi (Shiode *et al.*, 2002)

### 2.1.2 LBS Components

Menurut Wang (2008) Ada beberapa komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi, antara lain:

**a. *Mobile Devices***

Perangkat *Mobile* adalah salah satu komponen penting dalam LBS. Perangkat ini berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Perangkat *mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, *laptop*. Selain itu, perangkat *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

**b. *Communication Network***

Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari perangkat *mobile*-nya untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

**c. *Positioning Component***

Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah atau pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).

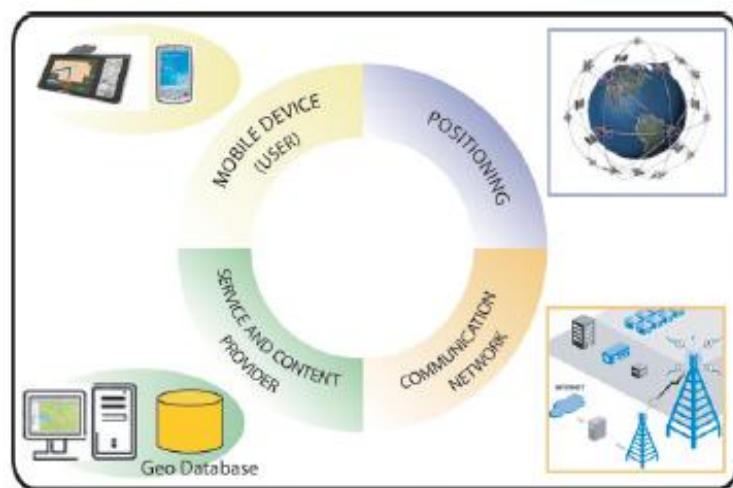
**d. *Service and Application Provider***

Penyedia layanan merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di Yellow Pages sesuai dengan permintaan, dan masih banyak lagi yang lainnya.

**e. *Data and Content Provider***

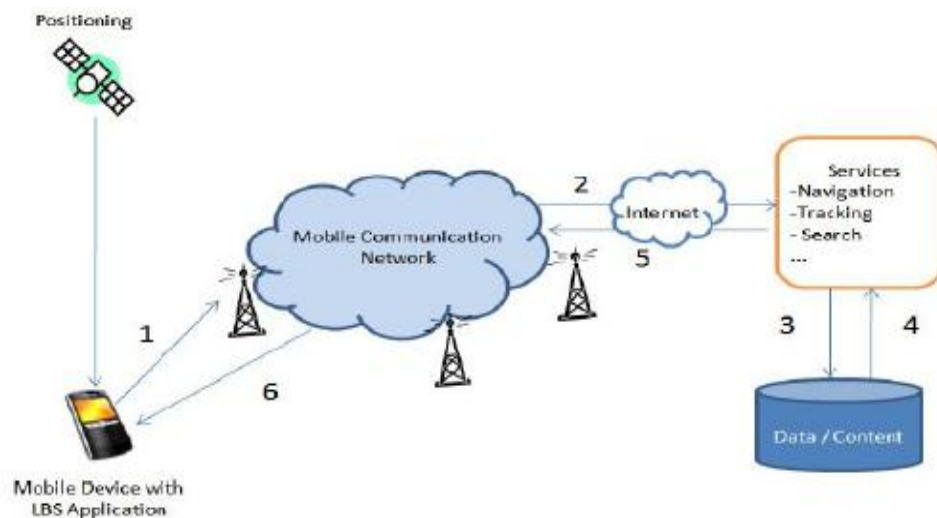
Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang atau pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan, bisnis, industri, dan bisa saja berasal dari Yellow Pages, maupun perusahaan penyedia data lainnya.

Secara lengkap komponen pendukung LBS tersebut dapat dilihat pada gambar berikut.



Gambar 2.2. Komponen dasar LBS (Wang, 2008)

Adapun gambar dibawah inimenunjukkan interaksi antar komponen dan proses LBS (Kushwaha, 2011).



Gambar 2.3. Komponen dan Proses LBS (Kushwaha, 2011)

Berikut penjelasan dari gambar 2.3 adalah:

**Langkah1** : Pengguna mengirimkan permintaan layanan menggunakan aplikasi yang berjalan pada perangkat mobile.

**Langkah2** : Permintaan layanan, dengan informasi lokasi pengguna saat ini diperoleh dari komponen posisi (dalam contoh ini, Data GPS), akan dikirim ke server layanan melalui jaringan komunikasi bergerak.

**Langkah 3, 4** : Permintaan server layanan basis data geografis dan database terkait lainnya untuk mendapatkan informasi yang dibutuhkan.

**Langkah 5, 6** : Akhirnya, informasi yang diminta dikirim kembali ke ponsel pengguna melalui jaringan komunikasi bergerak.

### 2.1.3 *Types of Location-Based Services*

Layanan berdasarkan lokasi perangkat memiliki kebutuhan akurasi yang berbeda, namun dapat dikategorikan oleh jenis layanan yang disediakan sebagai berikut:

#### a. *Emergency Services*

Kemampuan untuk menemukan individu yang tidak mengetahui lokasi dia yang sebenarnya atau tidak dapat mengungkapkan karena situasi

darurat. Hal ini memerlukan sistem yang secara otomatis menemukan lokasi pengguna *mobile* yang tepat setelah menerima panggilan darurat dan mentransfer informasi lokasi ke lembaga bantuan darurat.

**b. *Navigation Services***

Layanan navigasi didasarkan pada pengguna *mobile* yang memerlukan penunjuk arah lokasi geografis mereka. Kemampuan jaringan *mobile* untuk menemukan posisi yang tepat dari pengguna ponsel dapat diwujudkan dalam serangkaian navigasi berbasis jasa.

**c. *Information Services***

Layanan informasi secara umum merujuk pada distribusi digital dari konten perangkat terminal *mobile* berbasis lokasi mereka, kekhususan waktu dan perilaku pengguna. Pengguna *mobile* dapat memperoleh berbagai macam informasi lokasi seperti *landmark*, restoran, teater, dan pilihan transportasi umum.

**d. *Advertising Services***

iklan layanan berbasis lokasi berbentuk *banner mobile*, pesan SMS dan dipicu kedekatan iklan.

**e. *Tracking Services***

Aplikasi jenis ini juga dapat dimanfaatkan oleh perusahaan dalam rangka untuk mencari dan mengelola tim mereka bekerja, ini dikenal sebagai manajemen lapangan. Aplikasi lain adalah mereka digunakan untuk pelacakan produk dalam rantai pasokan.

**f. *Billing Services***

Lokasi penagihan yang strategis mengacu pada kemampuan penyedia layanan *mobile* untuk secara dinamis membebankan pengguna untuk layanan tertentu tergantung pada lokasi mereka ketika menggunakan atau mengakses layanan tersebut.

## **2.2 Algoritma Dijkstra**



Algoritma Dijkstra ditemukan oleh seorang ilmuwan komputer Belanda, Edsger Dijkstra pada tahun 1956 dan diterbitkan pada tahun 1959. Dijkstra merupakan Algoritma yang sangat terkenal untuk menyelesaikan permasalahan pencarian rute terpendek.

### **2.2.1 Definisi Algoritma Dijkstra**

Algoritma Dijkstra adalah algoritma penelusuran *graph* yang menyelesaikan permasalahan jalur terpendek dengan satu sumber asal untuk suatu graf dengan nilai sisi (*edge*) non negatif, menghasilkan pohon jalur terpendek (*shortest path tree*). Algoritma ini sering digunakan pada *routing* (<http://www.enotes.com>).

### **2.2.2 Pengenalan Algoritma Dijkstra**

Dijkstra menemukan jalur dengan biaya terendah (yaitu rute terpendek) antara simpul tersebut dengan setiap simpul lainnya. Algoritma ini juga dapat digunakan untuk menemukan jalur terpendek dari simpul asal ke simpul tujuan dengan cara menghentikan algoritma ketika jalur terpendek ke simpul tujuan telah ditentukan (Rutter, 2009).

Berikut ini adalah algoritma Dijkstra yang dijelaskan dalam bentuk notasi *pseudo code* sebagai berikut (Munir, 2005):

```

Procedur Dijkstra (input m: matriks, a : simpul awal)

( Mencari lintasan terpendek dari simpul awal a ke semua simpul lainnya
Masukan: matriks ketetanggaan (m) dari graf berbobot G dan simpul awal a
Keluaran: lintasan terpendek dari a ke semua simpul lainnya

}

Deklarasi

S1, S2, ..., Sn : integer{tabel integer}

d1, d2, ..., dn : integer{tabel integer}

i, j, k : integer

Algoritma

{ Langkah 0 (inisialisasi: }

for i ← 1 to n do

Si ← 0

di ← mai

endfor

{ Langkah 1: }

Sa ← 1 (karena simpul a adalah simpul asal lintasan terpendek, jadi
simpul a sudah pasti terpilih dalam lintasan terpendek)

da ← (tidak ada lintasan terpendek dari simpul a ke a)

{ Langkah 2, 3, ..., n - 1:}

for k ← 2 to n - 1 do

j ← simpul dengan Sj = 0 dan dj minimal

Sj ← 1 { simpul j sudah terpilih kedalam lintasan terpendek }

{ perbarui tabel d }

for semua simpul i dengan Si = 0 do

if dj + mji < di then

di ← dj + mji

endif

endfor

endfor

```

Cara kerja algoritma Dijkstra memakai strategi *greedy*. Dimana strategi *greedy* pada algoritma Dijkstra menyatakan bahwa pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih

dengan sebuah simpul lain yang belum terpilih. Lintasan dari simpul asal ke simpul yang baru haruslah merupakan lintasan yang terpendek diantara semua lintasannya ke simpul-simpul yang belum dipilih (Dewi, 2010).

Elemen-elemen penyusun prinsip Greedy pada Algoritma Dijkstra adalah (Faizah, 2010):

1. **Himpunan kandidat.** Himpunan ini berisi elemen-elemen yang memiliki peluang untuk membentuk solusi. Pada persoalan lintasan terpendek dalam graf, himpunan kandidat ini adalah himpunan simpul pada graf tersebut.
2. **Himpunan solusi.** Himpunan ini berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat namun tidak semuanya atau dengan kata lain himpunan solusi ini berupa bagian dari himpunan kandidat.
3. **Fungsi seleksi.** Fungsi seleksi adalah fungsi yang akan memilih setiap kandidat yang memungkinkan untuk menghasilkan solusi optimal pada setiap langkahnya.
4. **Fungsi kelayakan.** Fungsi kelayakan akan memeriksa apakah suatu kandidat yang telah terpilih (terseleksi) melanggar pembatas atau tidak. Apabila kandidat melanggar pembatas maka kandidat tidak akan dimasukkan ke dalam himpunan solusi.
5. **Fungsi objektif.** Fungsi objektif akan memaksimalkan atau meminimalkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi.

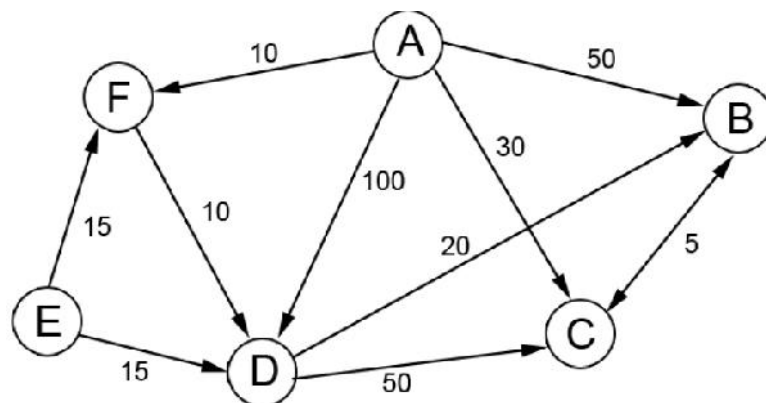
Ada beberapa kasus pencarian lintasan terpendek yang diselesaikan menggunakan algoritma Dijkstra (Faizah, 2010), yaitu:

- a. Pencarian lintasan terpendek antara dua buah simpul tertentu (*a pair shortestpath*).
- b. Pencarian lintasan terpendek antara semua pasangan simpul (*all pairs shortestpath*).
- c. Pencarian lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single-source shortest path*).

- d. Pencarian lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*).

Algoritma Dijkstra juga dapat digunakan untuk mencari lintasan terpendek dari sebuah titik yang ditentukan ke semua titik dalam gambar pada saat yang bersamaan, oleh sebab itu masalah tersebut seringkali disebut dengan *single-source shortest paths* problem. Masalah ini berhubungan dengan *spanning tree*. Grafik yang menggambarkan semua jalur dari titik yang satu ke semua titik yang lain pasti merupakan *spanning tree*, hal tersebut harus mencakup semua titik (Yulia dan Tanuhardja, 2002).

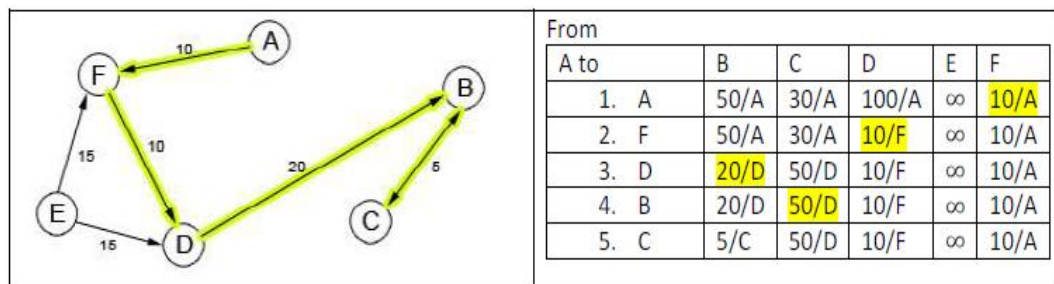
Lihat sebuah graf berarah pada Gambar 2.5, diberikan graf berbobot  $G=(V,E)$  dan akan ditelusuri graf dari titik A ke semua simpul lainnya yang dapat diakses. Dimana Titik A akan dipertimbangkan sebagai simpul asal. Sebelumnya, perhatikan bahwa graf yang akan digunakan adalah graf berbobot. Graf diberi bobot karena setiap sisinya harus bernilai non-negatif numerik. Pada masing-masing simpul mempunyai anak panah yang menunjukkan arah perjalanan yang memungkinkan simpul untuk dilalui. Adapun bobot untuk mencapai simpul B (50), C (30), D (100), dan F (10). Dimana bobot pada setiap sisi dapat menyatakan jarak, ongkos, waktu, dan sebagainya (Rutter, 2009).



Gambar 2.4. Graf Berarah (Rutter, 2009)

Dengan menerapkan algoritma Dijkstra diperoleh lintasan terpendek dengan jarak terpendek. Adapun hasil dari penelusuran graf berbobot dari simpul

asal ke simpul akhir adalah AFDBC dengan E tersis di tak terhitung, karena ujung dari anak panahnya hanya berasal dari simpul E dan jauh dari salah satu arah sehingga mustahil untuk mencapai E. Sebagaimana terlihat pada gambar dibawah ini.



Gambar 2.5. Hasil Penerapan Algoritma Dijkstra (Rutter, 2009)

## 2.3 Google Maps

Google Maps adalah layanan peta gratis Google yang cukup populer. Untuk menambahkan Google Maps kedalam website dapat dilakukan dengan menggunakan Google Maps API. Google Maps API dapat ditambahkan ke website menggunakan JavaScript. API tersebut menyediakan banyak fasilitas dan utilitas untuk memanipulasi peta dan menambahkan konten ke peta melalui berbagai layanan, memungkinkan untuk membuat aplikasi peta yang kuat pada website *user*.

Pengetahuan yang diperlukan untuk mengembangkan Google Maps API adalah tentang HTML dan JavaScript, sedangkan peta sudah disediakan oleh Google. Jadi developer hanya berkonsentrasi tentang data dan urusan peta ditangani oleh Google, sehingga dapat menghemat waktu. Agar peta dapat ditampilkan ke dalam website, maka diharuskan mempunyai *account* Google, kemudian mendaftarkannya di <http://code.google.com/apis/maps/signup.html>. Hal ini bertujuan untuk mendapatkan Google Maps API key (<http://www.code.google.com>, 2012).

## **2.4    *Object Oriented Analysis and Design (OOAD)***

*Object oriented analysis and design (OOAD)* merupakan suatu metode yang menuntun kita kepada suatu uraian yang berorientasi objek. Perancangan berorientasi objek mendefinisikan model dan proses untuk membangun sistem perangkat lunak yang kompleks, serta menawarkan seperangkat model logika dan fisik (Booch, 1998). Rational Software telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language (UML)* sebagai bahasa standar dalam *Object Oriented Analysis Design (OOAD)*.

### **2.4.1    *Unified Modelling Language (UML)***

*Unified Modelling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti dan Wahono, 2003).

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

UML mendefinisikan diagram-diagram sebagai berikut (Dharwiyanti dan Wahono, 2003):

#### **2.4.1.1 Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem (Dharwiyanti dan Wahono, 2003).

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

#### **2.4.1.2 Class Diagram**

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/ properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/ fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek

beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Dharwiyanti dan Wahono, 2003).

*Class* memiliki tiga area pokok:

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut:

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja

*Class* dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class*:

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.



4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

#### **2.4.1.3 Activity Diagram**

*Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum (Dharwiyanti dan Wahono, 2003).

#### **2.4.1.4 Statechart Diagram**

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring (Dharwiyanti dan Wahono, 2003).

#### **2.4.1.5 Sequence Diagram**

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequencediagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output*

tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan (Dharwiyanti dan Wahono, 2003).

#### **2.4.1.6 Deployment Diagram**

*Deployment/physical diagram* menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini (Dharwiyanti dan Wahono, 2003).

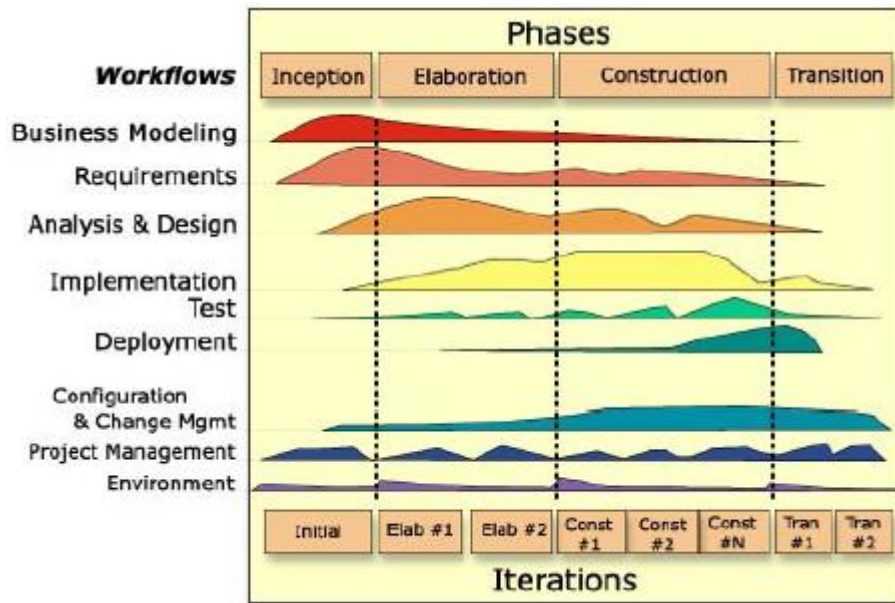
### **2.5 Rational Unified Process (RUP)**

Untuk pengembangan aplikasi LBS (*Location Based Services*) pada tugas akhir ini menggunakan metode pengembangan perangkat lunak *Rational Unified Process* (RUP).

#### **2.5.1 Definisi RUP**

*Rational Unified Process* (RUP) adalah sebuah Proses Rekayasa Perangkat Lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk memastikan perangkat lunak yang berkualitas tinggi dan sesuai kebutuhan penggunaanya dalam anggaran dan jadwal yang dapat diprediksi (Kruchten, 2001).

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Gambar 2.15 menunjukkan arsitektur keseluruhan dari RUP, dimana proses ini memiliki 2 dimensi, yaitu:



Gambar 2.6. Dua Dimensi RUP (Kruchten, 2001)

- **Dimensi pertama** digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.
- **Dimensi kedua** digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environtment*.

RUP mengarahkan kita terhadap pengembangan perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain (Kruchten, 2001):

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.
2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures* untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*,
6. *Control changes to software*. kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

### 2.5.2 Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception*, *Elaboration*, *Construction* dan *Transition* (Rational Team, 2001).

#### 1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi masalah, untuk itu diperlukan juga identifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambarannya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*. Hasil yang diperoleh pada fase ini adalah :

- a. Dokumen visi (visi dari kebutuhan proyek, kata kunci, batasan utama).
- b. Inisialisasi model *use-case* (10%-20% selesai).
- c. Daftar kata.
- d. *Business case*, termasuk didalamnya konteks bisnis, kriteria sukses, pengenalan pasar dan proyeksi keuangan.

- e. Inisialisasi penilaian resiko.
- f. Rencana proyek dan menunjukan fase serta iterasi.
- g. Model bisnis jika diperlukan

Kriteria evaluasi untuk fase *Inception* adalah :

- a. Menyesuaikan *stakeholder* dengan *scopedefinition* dan perkiraan biaya atau perkiraan jadwal.
- b. Pemahaman terhadap *use-case* utama
- c. Kredibilitas dari perkiraan biaya, jadwal, prioritas, resiko dan proses pengembangan.
- d. Pemahaman terhadap *prototype*

## 2. Fase *Elaboration*

Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-unsur yang memiliki resiko besar terhadap proyek. Adapun hasil dari fase *elaboration* adalah:

- a. *Use case* model, seluruh use case dan aktor telah teridentifikasi.
- b. *Requirement* tambahan yang mungkin tidak bersifat fungsional bagi proyek.
- c. *Software Architecture Description* (Deskripsi Arsitektur Perangkat Lunak).
- d. Prototipe dari arsitektur yang dapat dieksekusi.
- e. Revisi daftar tingkat resiko dan revisi *business-case*.
- f. Rencana pengembangan keseluruhan proyek.
- g. Persiapan dokumen panduan bagi pengguna (*user manual*).

Kriteria utama dalam fase *elaboration* melibatkan pertanyaan berikut :

- a. Apakah produk sudah stabil ?
- b. Apakah rancangan arsitekturalnya sudah stabil ?
- c. Apakah saat demo prototipe, unsur yang memiliki resiko telah bisa diatur ?
- d. Apakah rencana konstruksi telah detail dan akurat ?

- e. Apakah *stakeholder* bersedia dan menyepakati visi dari pengembangan proyek tersebut?
- f. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan dapat diterima?

### 3. Fase *Contruction*

Selama fase kontruksi, semua komponen dan fitur yang dikembangkan terintergrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase *Inception* dan *Elaboration*, menjadi pengembangan *deplyoableproduct*. Kriteria evaluasi terhadap fase *Construction* ini adalah :

- a. Apakah peluncuran produk cukup baik dan dapat diterima di komunitas pengguna?
- b. Apakah semua *stakeholder* siap untuk beralih ke komunitas pengguna?
- c. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

### 4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk. Kriteria evaluasi untuk fase *Transition* adalah :

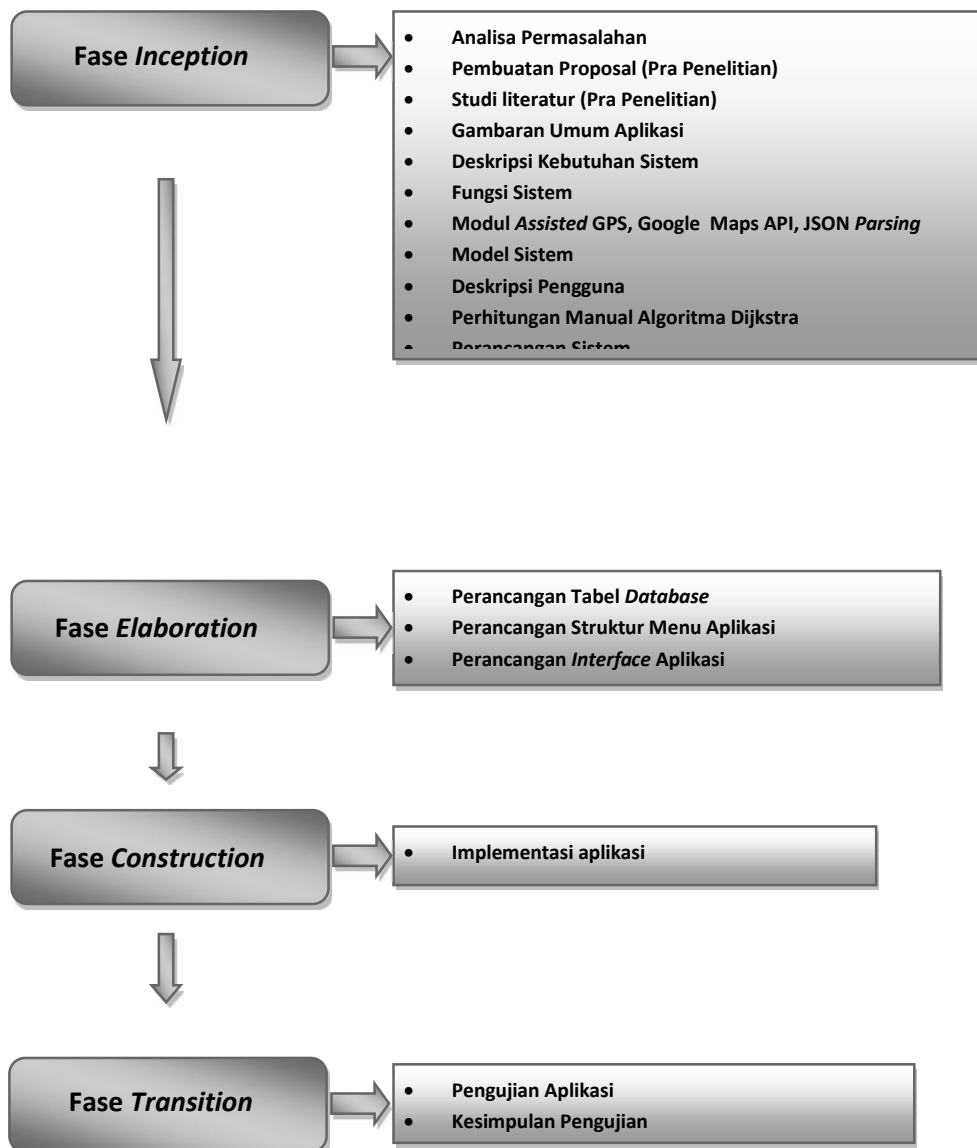
- a. Apakah pengguna merasa puas?
- b. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

### **bab iii**

## metodologi penelitian

### 3.1. Tahapan Penelitian

Tahapan penelitian yang akan dilaksanakan pada Tugas Akhir ini menggunakan metode *Rational Unified Process* (RUP), seperti yang telah dijelaskan pada bab landasan teori bahwa RUP merupakan suatu metode yang digunakan untuk proses pembangunan sebuah perangkat lunak. Gambar 3.1 di bawah ini menjelaskan tahapan penelitian terhadap aplikasi *location based service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra yang akan dibangun berdasarkan kepada metode RUP.





Gambar 3.1. Tahapan Penelitian dengan Metode RUP

### 3.2. Tahapan *Rational Unified Process* (RUP)

Berikut ini akan diuraikan tahapan-tahapan pembuatan aplikasi *Location Based Service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra, dengan menggunakan metode pengembangan *Rational Unified Process* (RUP).

#### 3.2.1. Fase *Inception*

Fase *Inception* merupakan fase untuk mengidentifikasi masalah yang ada pada aplikasi *location based service* (LBS) yang akan bangun, untuk itu diperlukan juga identifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambaranya. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Analisa permasalahan, yaitu memahami permasalahan yang terjadi, mengapa diperlukan aplikasi *location based service* (LBS).
- b. Pembuatan proposal untuk membangun aplikasi *location based service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra, yaitu mencakup latar belakang masalah, rumusan masalah, batasan masalah, dan tujuan penelitian.
- c. Studi literatur yang berhubungan dengan pembuatan aplikasi, mencakup penelusuran teori-teori yang berhubungan dengan permasalahan dan teknik membangun aplikasi *location based service* (LBS) untuk pencarian rute terpendek menggunakan algoritma Dijkstra.
- d. Gambaran umum aplikasi, yaitu menjelaskan secara umum tentang aplikasi yang akan dibangun.
- e. Deskripsi kebutuhan sistem, yaitu menjelaskan tentang kebutuhan yang diperlukan dalam membangun sistem.
- f. Fungsi sistem, bagian ini menjelaskan fungsi dari sistem yang akan dibangun.
- g. Modul *Assisted GPS*, *Google Maps API*, dan *JSON Parsing*.

- h. Model sistem, merupakan suatu gambaran dari cara kerja sistem.
- i. Deskripsi pengguna, bagian ini menjelaskan tentang pengguna sistem dan proses yang bisa dilakukan.
- j. Perhitungan manual algoritma Dijkstra, merupakan analisa perhitungan dan cara kerja algoritma Dijkstra yang dilakukan secara manual dengan suatu contoh kasus.
- k. Perancangan sistem, meliputi perancangan model dalam bentuk *Unified Modeling Language* (UML).

### **3.2.2. Fase *Elaboration***

Fase ini merupakan fase perancangan desain aplikasi yang akan dibangun, sesuai dengan hasil analisa pada fase sebelumnya. Pada fase ini akan dilakukan kegiatan sebagai berikut:

- a. Perancangan tabel *database*.
- b. Perancangan struktur menu aplikasi.
- c. Perancangan *interface* (antarmuka) aplikasi.

### **3.2.3. Fase *Construction***

Fase *Construction* merupakan tahapan membangun aplikasi LBS sesuai rencana yang telah dibuat pada tahap *elaboration*. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Implementasi aplikasi.

### **3.2.4. Fase *Transition***

Fase *Transition* merupakan tahap evaluasi dari aplikasi yang telah direncanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari aplikasi yang telah dibangun. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pengujian aplikasi.
- b. Kesimpulan pengujian.

## BAB IV

### ANALISA DAN PERANCANGAN

Bab ini merupakan bagian dari fase *inception* dan fase *elaboration* dari metode *Rational Unified Process* (RUP).

#### 4.1. Fase *Inception*

Pada bab sebelumnya telah dijelaskan tahapan-tahapan yang akan dilakukan pada fase *inception*. Pada bab analisa dan perancangan ini, fase *inception* hanya berfokus pada tahapan-tahapan penelitian yang akan dilakukan selanjutnya seperti yang akan dijelaskan berikut ini.

##### 4.1.1. Analisa Permasalahan

PT. CCAI merupakan perusahaan distributor minuman dalam kemasan, proses distribusi `pesanan *customer* dilakukan setiap hari kerja. Untuk wilayah Pekanbaru *customer* yang dimiliki berjumlah 2200, jumlah tersebut sudah termasuk *customer* yang aktif dan non aktif. Tersedianya banyak *customer* dapat dijadikan sebagai pilihan atau alternatif untuk menentukan rute *customer* yang akan dituju. Sistem kerja pada PT. CCAI yaitu dengan menggunakan pengetahuan karyawan sebagai pengendara mobil mengenai kota Pekanbaru dan alamat lokasi *customer* sebagai petunjuk, selanjutnya ditentukan urutan *customer* yang akan dikunjungi dari awal hingga akhir. Dari sistem kerja tersebut ditemukan suatu permasalahan, yaitu urutan *customer* yang diperoleh tidak mengacu pada suatu perhitungan yang akurat untuk masing-masing jarak antar *customer*, sehingga waktu yang diperoleh tidak efisien dan belum optimal.

Pekanbaru merupakan kota yang terdiri dari banyak jalan yang dapat diakses penggunaannya untuk mencapai suatu tujuan dan masing-masing jalan tersebut memiliki ukuran panjang tertentu dan berbeda-beda. Disamping itu juga terdapat rambu-rambu lalu lintas yang dapat digunakan pengendara sebagai panduan dalam menentukan jalan satu arah dan jalan yang tidak bisa dilewati.

Aplikasi LBS yang akan dibangun menggunakan algoritma Dijkstra sebagai metode pencarian rute terpendek. Kriteria yang digunakan dalam menentukan rute terpendek adalah jarak garis lurus antar lokasi berdasarkan perhitungan dari Google Maps. Aplikasi ini juga menggunakan teknologi Google Maps sebagai penunjuk rute jalan dari lokasi awal menuju ke lokasi tujuan. Fitur yang akan dimiliki Google Maps pada aplikasi berdasarkan perkembangan teknologinya, pada saat ini Google Maps yang digunakan pada aplikasi ini belum bisa menampilkan jalan satu arah dan jalan yang tidak bisa dilewati, hal ini disebabkan Google Maps belum memfasilitasi fitur tersebut, sehingga diperlukan juga pertimbangan pengendara mobil antara rute yang ditampilkan aplikasi dengan rambu-rambu atau aturan-aturan lalu lintas yang berlaku.

Data yang diperlukan dalam proses perhitungan jarak antar *customer* yaitu data koordinat dari masing-masing *customer*. Data koordinat tersebut dapat diperoleh dengan mengunjungi lokasi *customer* dan menggunakan perangkat yang dilengkapi dengan teknologi *Global Positioning System* (GPS), hal ini diperlukan agar titik koordinat yang diperoleh tersebut lebih akurat. Adapun data pendukung lainnya seperti nama *customer*, alamat, dan foto.

Adapun teknologi yang dibutuhkan adalah berupa perangkat *mobile* untuk menjalankan aplikasi ini yaitu *smartphone* Android yang memiliki sistem operasi minimal Android versi 2.2 (*Froyo*).

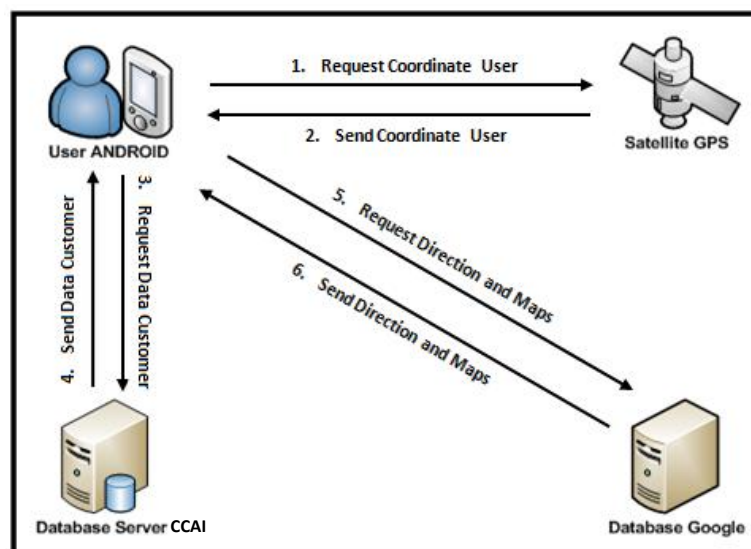
#### **4.1.2. Gambaran Umum Aplikasi**

Aplikasi *location based services* (LBS) yang akan dibangun merupakan aplikasi yang berbasis *client-server*, dan akan dijalankan pada perangkat *mobile* Android dengan teknologi *Global Positioning System* (GPS) *receiver*. Aplikasi LBS ini memiliki fitur layanan navigasi (*navigation services*), sebagai penunjuk arah lokasi geografis pengguna. Aplikasi menggunakan *database server* dan bahasa pemrograman PHP sebagai *connector* dalam pengiriman *request* dan penerimaan respon terhadap *server* yang menggunakan Apache dan *database* MySQL. Aplikasi *location based services* (LBS) ini dapat digunakan untuk mengetahui informasi seperti daftar urutan pelanggan yang akan dikunjungi dan

route jalan untuk menuju lokasi *customer*. Aplikasi ini diberi nama CCAI (LBS for PT. CCAI), sehingga untuk selanjutnya penyebutan aplikasi adalah CCAI.

CCAI dititik beratkan pada pengambilan titik koordinat bumi perangkat *mobile*, dalam pengambilan koordinat CCAI memanfaatkan teknologi GPS. Kemudian untuk *request* peta dan rute, CCAI menggunakan Google Maps API dengan memberikan parameter koordinat bumi. Setelah mengirim parameter tersebut ke Google Maps, maka Google Maps *server* akan membalas berupa peta statik (gambar). Dalam pencarian rute, CCAI mengirimkan dua koordinat bumi sebagai titik awal dan titik tujuan, kemudian Google Maps *server* akan membalas berupa data rute dalam bentuk *JavaScript Object Notation* (JSON) yang selanjutnya data tersebut akan di*Parsing* oleh CCAI dan ditampilkan kepada pengguna, sehingga aplikasi dapat menampilkan rute jalan untuk menuju lokasi *customer*.

Gambaran umum aplikasi CCAI inibertujuan untuk memberikan gambaran mengenai struktur menu dan konsep dasar aplikasi. Untuk lebih jelasnya deskripsi arsitektur sistem ini dapat di lihat pada gambar 4.1 dibawah ini.



Gambar 4.1. Arsitektur Sistem

Berikut penjelasan dari gambar 4.1 diatas tentang deskripsi arsitektur sistem yang akan dibangun:

1. Perangkat Android merupakan perangkat tempat berjalannya CCAI. Dari perangkat inilah pengguna berinteraksi dengan CCAI, dan melakukan *request* data koordinat kepada satellite GPS, selanjutnya pengguna mendapatkan titik koordinat lokasi awal.
2. Pengguna melakukan *request* data yang dibutuhkan ke *database server*, selanjutnya *database server* akan mengirimkan data tersebut ke perangkat android melalui CCAI.
3. Pengguna mengirimkan koordinat perangkat sebagai titik awal dan koordinat pelanggan sebagai titik tujuan ke Google Maps *server* untuk memperoleh *Maps* dan *Direction*. Selanjutnya Google Maps *server* akan menampilkan hasil dari *request* tersebut pada perangkat android.

Data dan informasi yang dibutuhkan pengguna CCAI dapat diakses dimana saja selama perangkat Android tersebut terhubung dengan jaringan internet dan masih dalam wilayah kota Pekanbaru.

#### **4.1.3. Deskripsi Kebutuhan Sistem**

Kebutuhan sistem merupakan hal yang harus diperhatikan. Sistem yang akan dibangun membutuhkan beberapa komponen pendukung, sehingga sistem dapat dibangun sesuai dengan tujuan yang diharapkan. Berikut ini adalah deskripsi kebutuhan sistem.

##### **4.1.3.1. Sistem yang akan dibangun**

Untuk kebutuhan sistem yang akan dibangun terdapat dua bagian, pertama kebutuhan aplikasi CCAI dari perangkat android. Kedua *connector* menggunakan bahasa pemrograman PHP sebagai penghubung antara CCAI di perangkat Android dengan *database server*, dimana *database server* digunakan sebagai media penyimpanan data dan informasi yang dibutuhkan CCAI.

##### **a. Aplikasi Pada Perangkat Android**

Kebutuhan aplikasi pada perangkat android adalah:

1. Bahasa pemrograman yang digunakan adalah *Java*.
2. Aplikasi menampilkan proses *login* untuk mengakses aplikasi.
3. Aplikasi menampilkan daftar *customer* berdasarkan proses algoritma Dijkstra. Apabila diklik salah satu *customer*, maka selanjutnya aplikasi akan menampilkan informasi *customer* berupa foto lokasi *customer*, nama *customer*, alamat.
4. Aplikasi ini dapat menampilkan map *customer* sesuai dengan nama *user*. Apabila diklik salah satu *customer* yang terdapat dimap, maka aplikasi akan menampilkan informasi berupa nama *customer* dan alamat.
5. Aplikasi ini juga dapat menampilkan map untuk mengetahui rute jalan menuju lokasi *customer* yang akan ditampilkan oleh Google Maps.

**b. Connector**

*Connector* ini berfungsi sebagai perantara atau penghubung antara aplikasi CCAI yang berjalan di perangkat Android dengan *database server*. *Connector* dibangun menggunakan bahasa pemrograman PHP. Berikut beberapa Mesin yang akan dibangun:

1. *Login connector*, berfungsi menangani *request* login pengguna dari aplikasi pada perangkat Android dan meneruskan ke *database*. Kemudian mengembalikan respon dari *database* untuk diteruskan ke aplikasi di Android, dan ditampilkan pada halaman *login*.
2. *CustomerListconnector*, berfungsi menangani *request* daftar *customer* dari aplikasi pada perangkat Android dan meneruskan ke *database*. Kemudian mengembalikan respon dari *database* untuk diteruskan ke aplikasi di Android, dan ditampilkan pada halaman daftar *customer*.
3. *CustomerDetail connector*, berfungsi menangani *request* informasi lengkap *customer* dari aplikasi pada perangkat Android dan meneruskan ke *database*. Kemudian mengembalikan respon dari *database* untuk diteruskan ke aplikasi di Android, dan ditampilkan pada halaman detail *customer*.

#### 4.1.3.2. AnalisisFungsional

Analisis fungsional merupakan penjelasan mengenai fitur-fitur yang akan ada pada aplikasi CCAI. Fitur-fitur tersebut antara lain:

1. Pengguna diizinkan melakukan proses *login* untuk mengakses aplikasi.
2. Pengguna diizinkan melihat daftar *customer*.
3. Pengguna diberikan hak akses untuk melihat detail *customer*.
4. Pengguna diizinkan melihat map yang menampilkan titik lokasi *customer*.
5. Pengguna dapat melihat informasi rute jalan menuju lokasi *customer* melalui map.

#### 4.1.3.3.Performansi Aplikasi

Aplikasi CCAI merupakan aplikasi yang berjalan dilingkungan perangkat bersistem operasi Android. Terdapat beberapa keterbatasan yang ditemui pada perangkat ini, sehingga perlu diperhatikan untuk menjadi acuan dalam pengembangan aplikasi ini, yaitu diantaranya:

1. Sumber daya yang terbatas, hingga saat ini perangkat Android yang banyak beredar memiliki kapasitas memori terbatas.
2. Sumber daya baterai yang secara efektif hanya mampu bertahan selama kurang lebih 200 jam dalam keadaan *standby*.

Dari keterbatasan-keterbatasan pada perangkat Android, maka diusulkan beberapa alternatif untuk meningkatkan performa aplikasi terhadap keterbatasan yang ada, diantaranya:

1. Merancang aplikasi yang menggunakan memori seefektif mungkin, sehingga tidak mengganggu siklus operasi Android dan aplikasi lain.
2. Merancang aplikasi dengan pemanfaatan sumber daya seefisien mungkin namun tidak mengurangi fungsi dan performa aplikasi.
3. Merancang aplikasi dengan antarmuka yang sederhana namun tetap menarik dan ramah bagi pengguna.



#### **4.1.4. Fungsi Sistem**

Secara umum fungsi sistem ada dua bagian yaitu aplikasi yang akan dibangun dari sisi perangkat Android dan *databaseserver*.

##### **4.1.4.1.Fungsi Aplikasi dari Sisi Perangkat Android**

Aplikasi yang akan dibangun dari sisi perangkat Android memiliki fungsi-fungsi sebagai berikut:

1. Menampilkan proses *login*.
2. Menampilkan daftar *customer*.
3. Menampilkan map beserta titik koordinat yang telah ditandai.
4. Menampilkan informasi koordinat *customer*.
5. Menampilkan detail *customer*.
6. Menampilkan map untuk memperoleh rute jalan menuju lokasi *customer*.

##### **4.1.4.2.Fungsi Database Server**

*Databaseserver* memiliki fungsi-fungsi sebagai berikut :

1. Berfungsi mengelola data pengguna.
2. Berfungsi mengelola data *customer*.

#### **4.1.5. Modul Assisted GPS, Modul Google Maps API, dan Modul JSON Parsing**

Modul-modul yang akan dijelaskan adalah sekilas mengenai teknologi yang digunakan pada saat membangun sistem, diantaranya:

##### **4.1.5.1. Modul Assisted GPS**

Pada aplikasi CCAI, teknologi *Location Based Service* (LBS) sangat dibutuhkan untuk mengetahui posisi pengguna dengan memanfaatkan *Global Positioning System* (GPS) yang terintegrasi dengan perangkat Android. Satellite GPS akan memberitahu posisi berupa koordinat bumi yang dikenal dengan *latitude* dan *longitude*.

#### 4.1.5.2. Modul Google Maps API

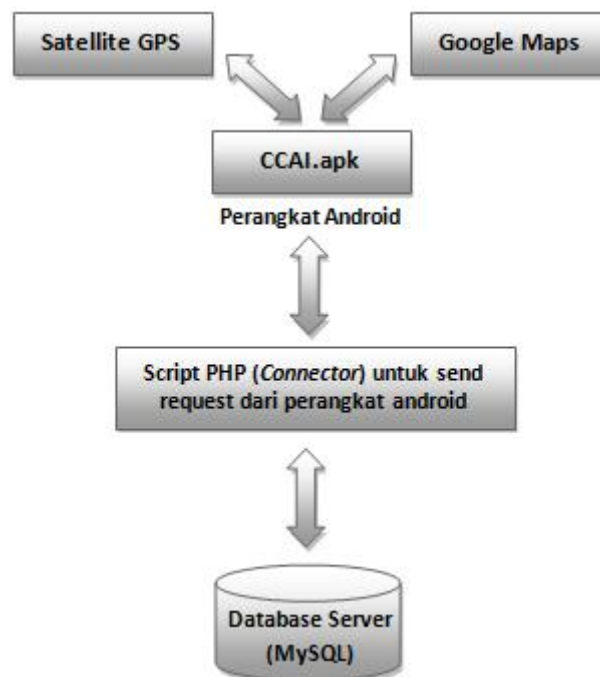
Salah satu fitur dari aplikasi CCAI adalah menampilkan map yang dapat digunakan sebagai acuan pengguna dalam mencari rute jalan dan letak lokasi pelanggan. Menurut penulis *map* yang paling tepat untuk digunakan dalam memperoleh informasi tersebut adalah Google Maps, dikarenakan teknologi Google Maps yang sangat berkembang saat ini, juga sesuai dengan perangkat android yang digunakan.

#### 4.1.5.3. Modul JSON Parsing

Pada aplikasi CCAI ini menggunakan *JavaScript Object Notation* (JSON) *Parsing* untuk membuat rute pada map.

#### 4.1.6. Model Sistem

Model ini dirumuskan sebagai fungsi yang menggambarkan hubungan antar objek-objek yang berperan dalam sistem.



Gambar 4.2. Model Sistem

Objek-objek yang berperan pada model sistem diatas, yaitu :

1. Satellite GPS digunakan untuk memperoleh koordinat lokasi.
2. Google Maps sebagai peta yang digunakan pada aplikasi CCAI.

3. Perangkat Android sebagai media untuk mengakses aplikasi CCAI dengan hak akses sebagai pengguna, di dalamnya dipasang CCAI.apk yang dibuat. Fungsi yang dapat dilakukan oleh perangkat Android adalah menampilkan *login* pengguna, informasi *customer*, map, dan *direction*.
4. *Script* PHP untuk *request* dari perangkat Android sebagai antarmuka antara perangkat Android dengan *database*.
5. *Database* yang menggunakan MySQL sebagai tempat penyimpanan data yang digunakan untuk pemrosesan pada aplikasi CCAI.

#### 4.1.7. Deskripsi Pengguna

Pengguna dari aplikasi ini adalah supir (*driver*), supir diberikan akses penuh terhadap semua fitur dan fungsi yang ada pada aplikasi ini. Untuk lebih jelasnya dapat dilihat di tabel 4.1 di bawah ini.

Tabel 4.1. Deskripsi Pengguna

No	Kategori Pengguna	Hak Akses	Keterangan
1	Admin (Database Server CCAI)	Mengelola <i>database user</i> dan <i>database customer</i>	Hak akses penuh
2	<i>User</i> (Perangkat Android)	a. Melakukan <i>Login</i> b. Melihat <i>customer list</i> c. Melihat maps d. Melihat Informasi pelanggan Melihat rute menuju lokasi pelanggan	Hak akses penuh

#### 4.1.8. Perhitungan Manual Algoritma Dijkstra

Input algoritma ini adalah sebuah graf berarah yang berbobot (*weighted directed graph*) dan sebuah simpul asal dalam G. Setiap sisi dari graf

ini adalah pasangan simpul  $(u,v)$  yang melambangkan hubungan dari simpul  $u$  ke simpul  $v$ .

$G = (V, E)$

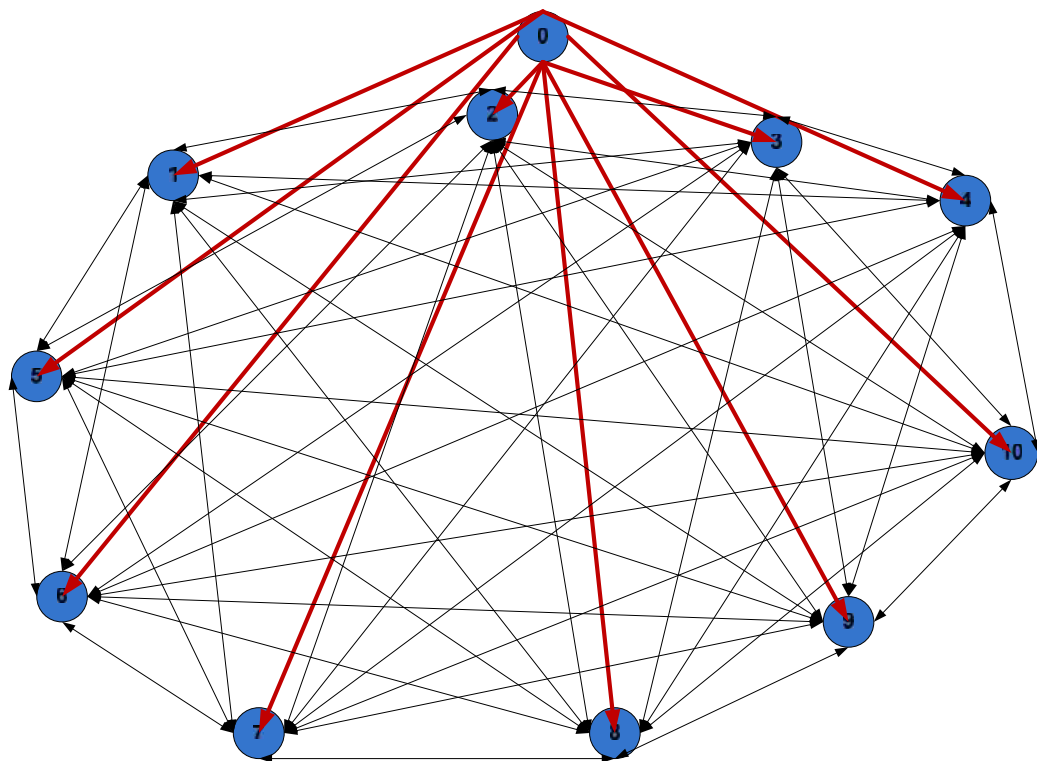
Dimana :  $V$  = Simpul,  $E$  = Sisi

Algoritma Dijkstra memakai 2 set dari simpul:

$S$  = Simpul asal

$V$  = Himpunan semua simpul

Jadi  $(u,v)$  adalah jarak tak-negatif dari simpul  $u$  ke simpul  $v$ . Berikut gambar sebuah graf berbobot  $G = (V, E)$  dan sebuah simpul awal 0. Berdasarkan kasus di PT. CCAI, dimana satu mobil atau supir menangani 10 pelanggan, selanjutnya dimulai dari posisi awal 0 yaitu PT. CCAI akan dilakukan perjalanan ke semua lokasi *customer*. Untuk lebih lengkapnya dapat dilihat pada gambar 4.3.




Gambar 4.3. Graf Berbobot

Keterangan :

— ● Simpul

–  Satu Arah

–  Dua Arah

- |                     |                    |                     |
|---------------------|--------------------|---------------------|
| 0. PT. CCAI         | 4. Amazone SKA     | 8. Aziziah Swalayan |
| 1. Elang Futsal     | 5. Hotel Dharma U. | 9. Inter. Club MP   |
| 2. Al Barokah Mart  | 6. Kevinn Mart     | 10. Central Pa      |
| 3. Mini Mart Adelia | 7. AA Mart         |                     |

Tabel 4.2. Jarak antar PT. CCAI dan *Customer* (Km)

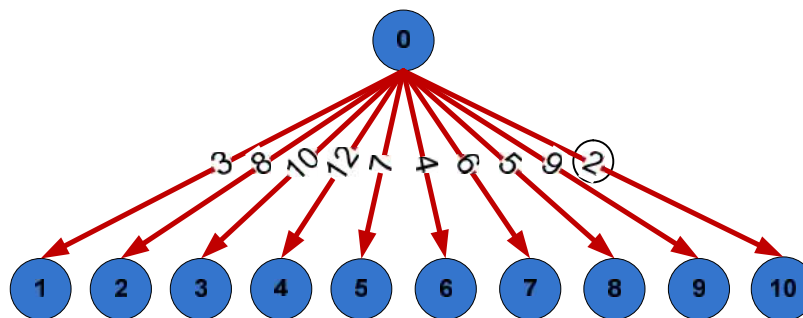
		SIMPUL TUJUAN										
		0	1	2	3	4	5	6	7	8	9	10
SIMPUL AWAL	0	0	3	8	10	12	7	4	6	5	9	2
	1		0	9	7	4	1	2	6	3	8	5
	2		9	0	5	8	10	3	1	7	4	3
	3		7	5	0	5	6	6	4	2	1	8
	4		4	8	5	0	3	7	2	12	10	1
	5		1	10	6	3	0	5	10	4	2	10
	6		2	3	6	7	5	0	2	4	8	6
	7		6	1	4	2	10	2	0	9	7	12
	8		3	7	2	12	4	4	9	0	3	2
	9		8	4	1	10	2	8	7	3	0	4
	10		5	3	8	1	10	6	12	2	4	0

Algoritma Dijkstra akan melakukan pengembangan pencarian dari satu simpul awal ke simpul lainnya didalam graf dan ke simpul selanjutnya tahap demi tahap. Berikut ini urutan logika dari algoritma Dijkstra:

1. Set 0 sebagai simpul awal ( $s$ ), simpul 0 terhubung dengan simpul lainnya dan memiliki bobot (jarak) masing-masing.
2. Set simpul awal sebagai “Simpul Keberangkatan” dan set semua simpul lainnya sebagai “Belum Terjamah”.
3. Dari simpul keberangkatan ( $s$ ), pertimbangkan simpul tetangga ( $v$ ) yang belum terjamah dan hitung jaraknya dari simpul keberangkatan.
4. Ketika telah selesai mempertimbangkan setiap jarak terhadap simpul tetangga, tandai simpul yang telah terjamah sebagai “Simpul Terjamah”. Simpul terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “Belum terjamah” dengan bobot minimal (dari simpul keberangkatan) sebagai “Simpul Keberangkatan” ( $u$ ) selanjutnya, dan lakukan pengulangan dengan kembali ke *step* 3 hingga simpul terakhir ditemukan dan tidak ada lagi simpul belum terjamah.

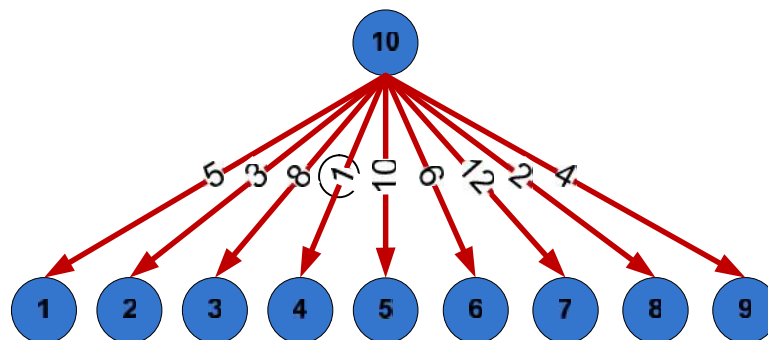
Selanjutnya penjelasan langkah per langkah pencarian rute terpendek secara rinci dimulai dari simpul awal 0 sampai ke simpul tujuan dengan nilai jarak minimum.

1. Simpul awal 0, dan simpul tujuan 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Setiap jarak yang menghubungkan simpul memiliki nilai masing-masing. Algoritma Dijkstra melakukan kalkulasi terhadap simpul tetangga yang terhubung langsung dengan simpul keberangkatan (simpul 0), dan hasil yang didapat adalah simpul 10, karena bobot nilai simpul 10 paling kecil jika dibandingkan dengan nilai pada simpul lain, nilai 2 Km.



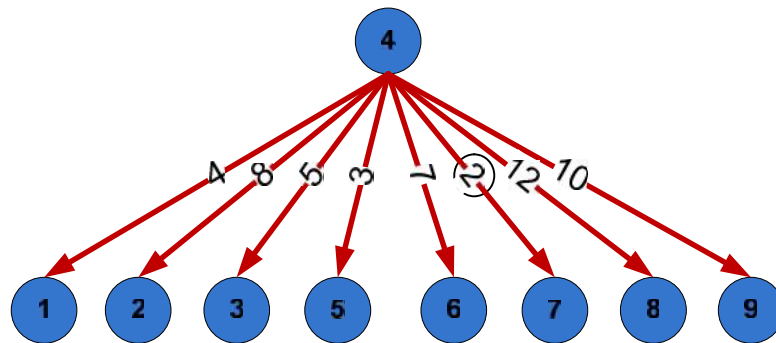
Gambar 4.4. Langkah 1

2. Simpul 10 diset menjadi simpul keberangkatan dan ditandai sebagai simpul terjamah. Dijkstra melakukan kalkulasi kembali terhadap simpul-simpul tetangga yang terhubung langsung dengan simpul yang telah terjamah, dan kalkulasi Dijkstra menunjukkan bahwa simpul 4 yang menjadi simpul keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 1 Km.



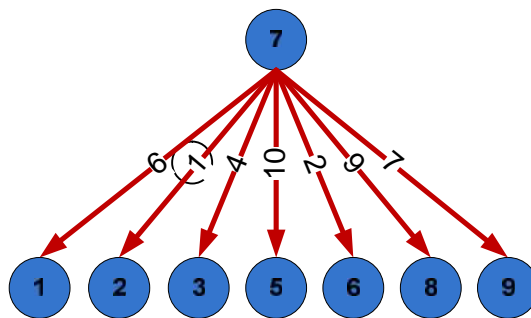
Gambar 4.5. Langkah 2

3. Perhitungan berlanjut dengan simpul 4 ditandai menjadi simpul yang telah terjamah. Dari semua simpul tetangga belum terjamah yang terhubung langsung dengan simpul terjamah, simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 7 karena nilai bobot yang terkecil, nilai 2 Km.



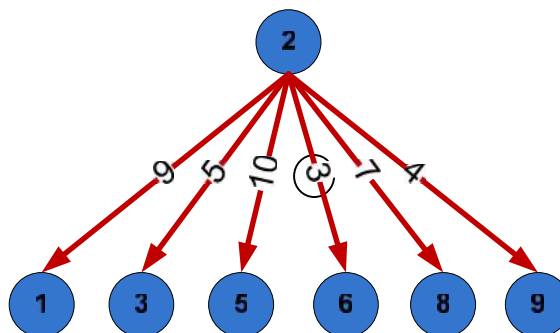
Gambar 4.6. Langkah 3

4. Kemudian dilanjutkan lagi dengan simpul 7 ditandai menjadi simpul yang telah terjamah. Simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 2 karena nilai bobot yang terkecil, nilai 1 Km.



Gambar 4.7. Langkah 4

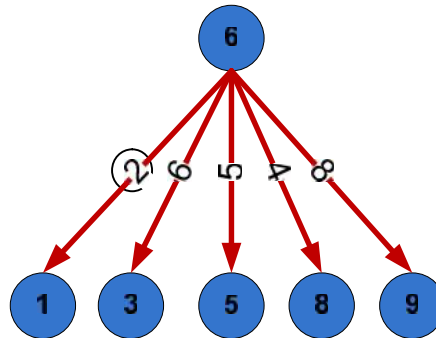
5. Selanjutnya dengan simpul 2 ditandai menjadi simpul yang telah terjamah. Simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 6 karena nilai bobot yang terkecil, nilai 3 Km.





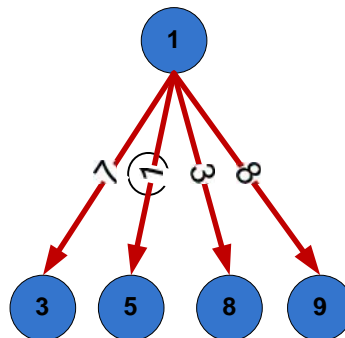
Gambar 4.8. Langkah 5

6. Simpul 6 ditandai menjadi simpul yang telah terjamah. Simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 1 karena nilai bobot yang terkecil, nilai 2 Km.



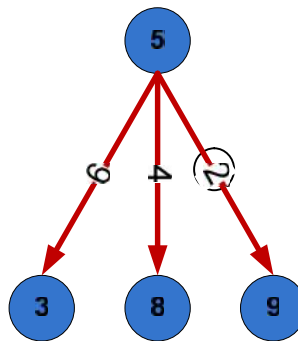
Gambar 4.9. Langkah 6

7. Simpul 1 ditandai menjadi simpul yang telah terjamah. Simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 5 karena nilai bobot yang terkecil, nilai 1 Km.



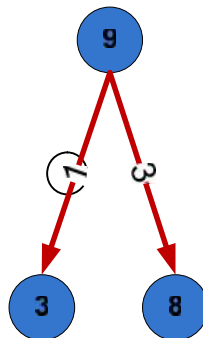
Gambar 4.10. Langkah 7

8. Simpul 5 ditandai menjadi simpul yang telah terjamah. Simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 9 karena nilai bobot yang terkecil, nilai 2 Km.



Gambar 4.11. Langkah 8

9. Simpul 9 ditandai menjadi simpul yang telah terjamah. Simpul selanjutnya yang ditandai menjadi simpul terjamah adalah simpul 3 karena nilai bobot yang terkecil, nilai 1 Km.



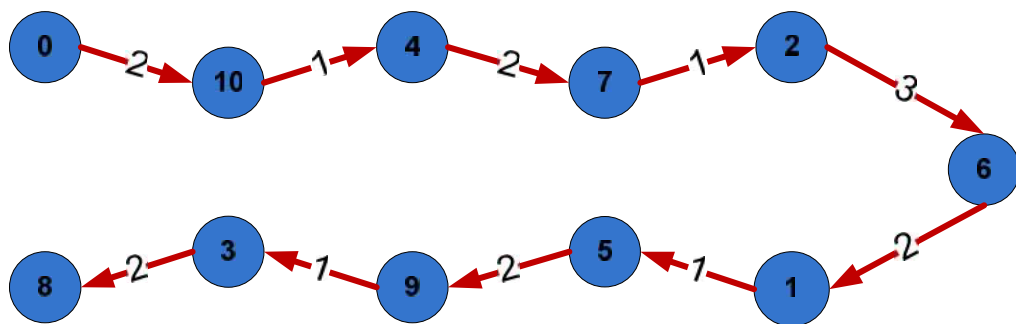
Gambar 4.12. Langkah 9

10. Simpul 3 menjadi simpul terjamah, Dijkstra melakukan kalkulasi kembali, dan menemukan bahwa simpul 8 sebagai simpul terakhir telah tercapai. Bila simpul tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.



Gambar 4.13. Langkah 10

Jadi rute terpendeknya adalah 0-10-4-7-2-6-1-5-9-3-8, dan nilai bobot yang didapat adalah 17 Km.



Gambar 4.14. Rute Terpendek

#### 4.1.9. Perancangan Sistem

Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan aplikasi CCAIberbasis *client-server* pada sistem operasi Android. Perancangan-perancangan yang akan dijelaskan dalam

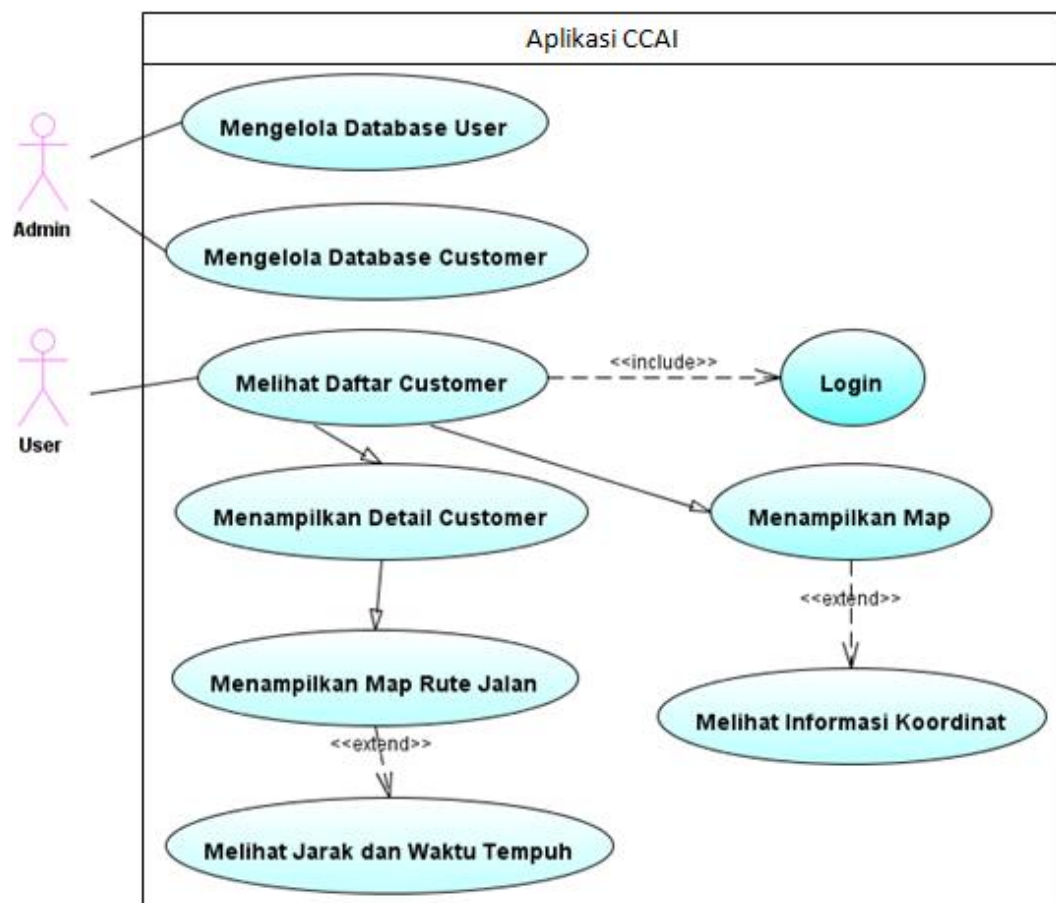
laporan ini meliputi perancangan model dalam bentuk *Unified Modeling Language* (UML) yang terdiri dari *Usecase Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Selain itu juga ada perancangan hubungan antar *class* di Android dengan modul penghubung dan perancangan *interface* sistem yang terdiri dari perancangan *prototype* dan struktur menu.

#### 4.1.9.1. Pemodelan UML (*Unified Modeling Language*)

Analisa perancangan pada sistem ini menggunakan UML, yaitu terdiri dari *usecase diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*.

##### 4.1.9.1.1. *Usecase Diagram*

*Usecase diagram* merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor dan sistem. Gambar 4.15 dibawah ini menjelaskan aliran *usecase diagram*.



Gambar 4.15. Aliran *Usecase Diagram*

#### 4.1.9.1.2. *Use Case Specification*

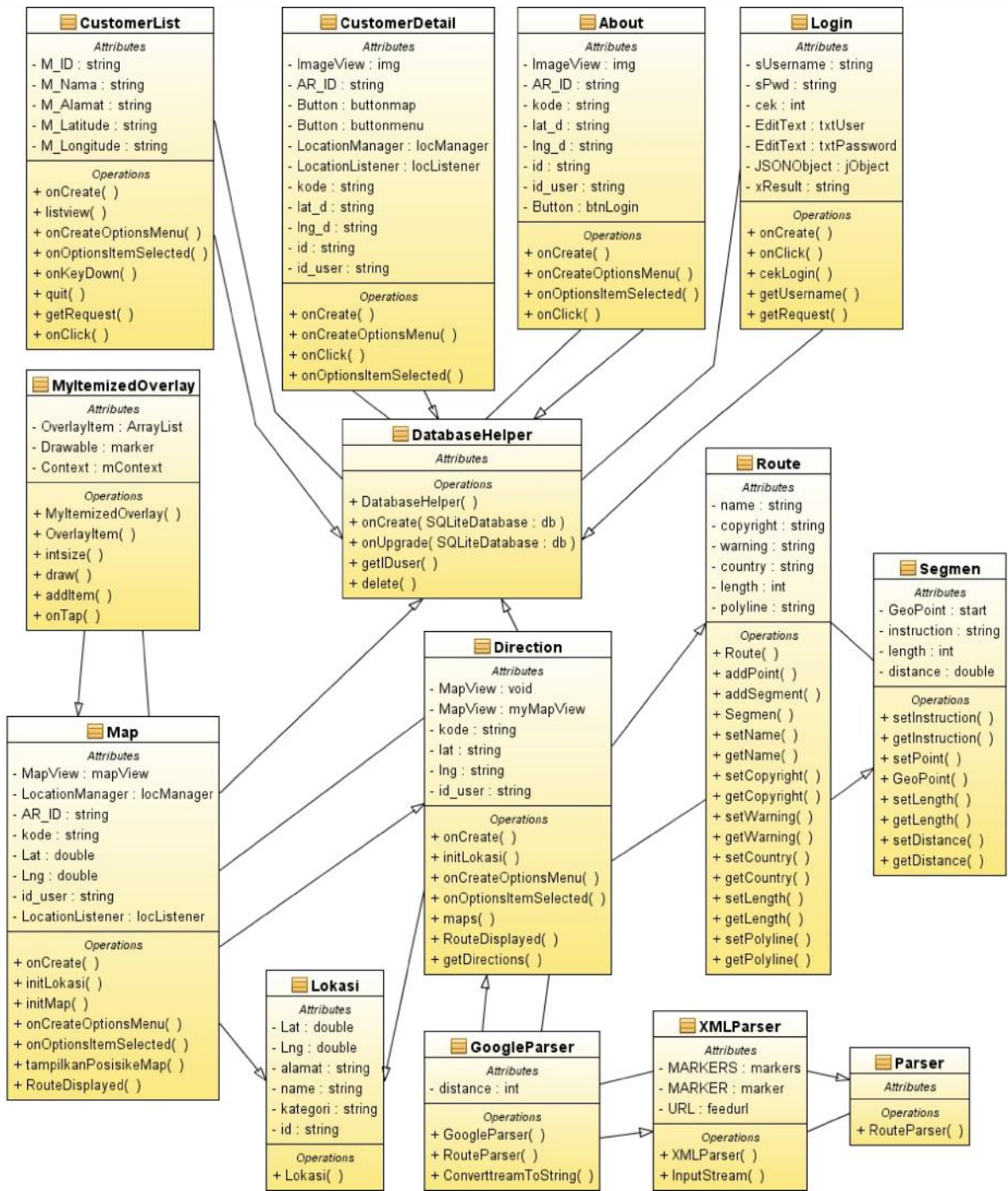
Untuk menjelaskan alur dari gambar *use case diagram*, maka selanjutnya dideskripsikan di *use case specification* melihat daftar *customer* yang dapat dilihat pada Tabel 4.3. Sedangkan *use case specification* lainnya dapat dilihat pada Lampiran A.

Tabel 4.3. *Use Case Specification* Melihat Daftar *Customer*

Nama <i>Use Case</i>	Melihat Daftar <i>Customer</i>
Deskripsi Singkat	<i>User</i> melihat halaman daftar <i>customer</i> yang ditampilkan aplikasi untuk melihat <i>customer</i> yang akan dikunjungi
Aktor	<i>User</i>
<i>Pre Condition</i>	<i>User</i> harus melakukan login untuk mengakses aplikasi
<i>Post Condition</i>	Aplikasi menampilkan halaman daftar <i>customer</i>
<i>Normal Flow</i>	<i>User</i> melakukan <i>login</i> untuk mengakses aplikasi kemudian aplikasi akan menampilkan daftar <i>customer</i>
<i>Alternate Flow</i>	Seandainya <i>user</i> ingin melihat detail <i>customer</i> maka diklik salah satu nama <i>customer</i>

#### 4.1.9.1.3. *Class Diagram*

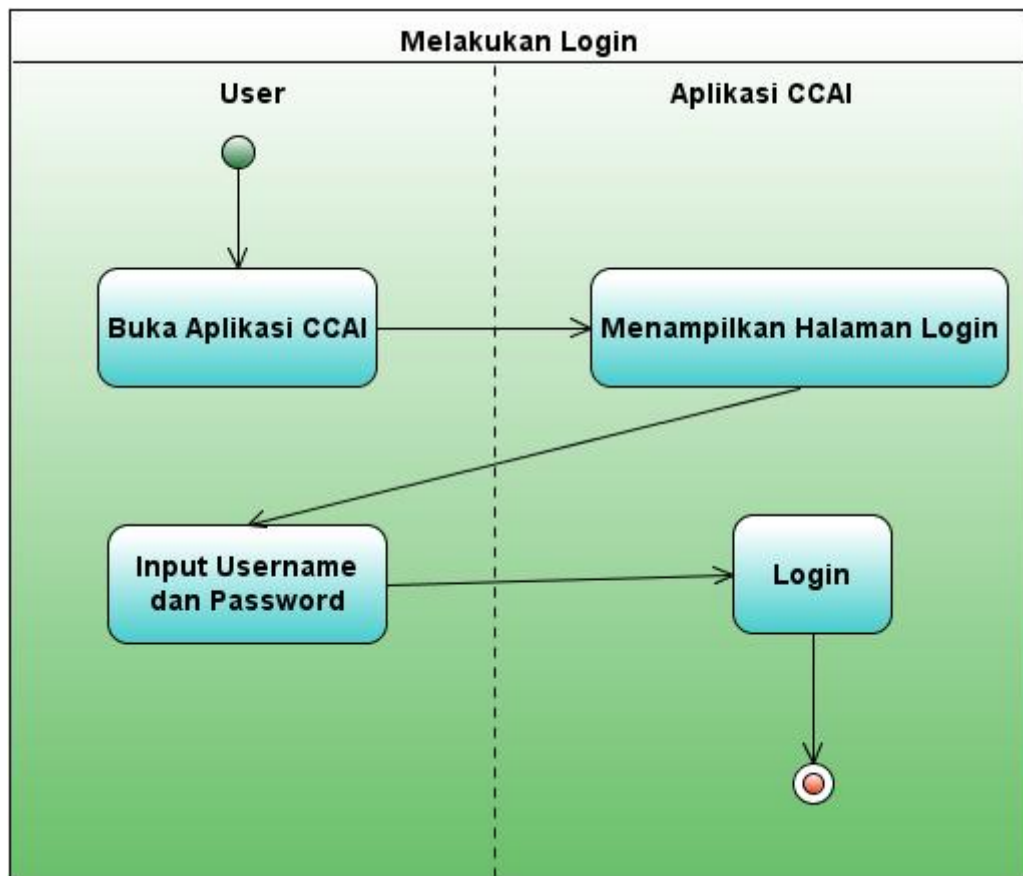
*Class diagram* digunakan untuk menggambarkan hubungan antara kelas yang ada pada aplikasi CCAI. Penggambaran *class diagram* berikut ini merupakan *class diagram* sistem yang terdapat pada perangkat android. Untuk melihat *class diagram* dapat dilihat pada gambar 4.16.



Gambar 4.16. *Class Diagram*

#### **4.1.9.1.4. *Activity Diagram***

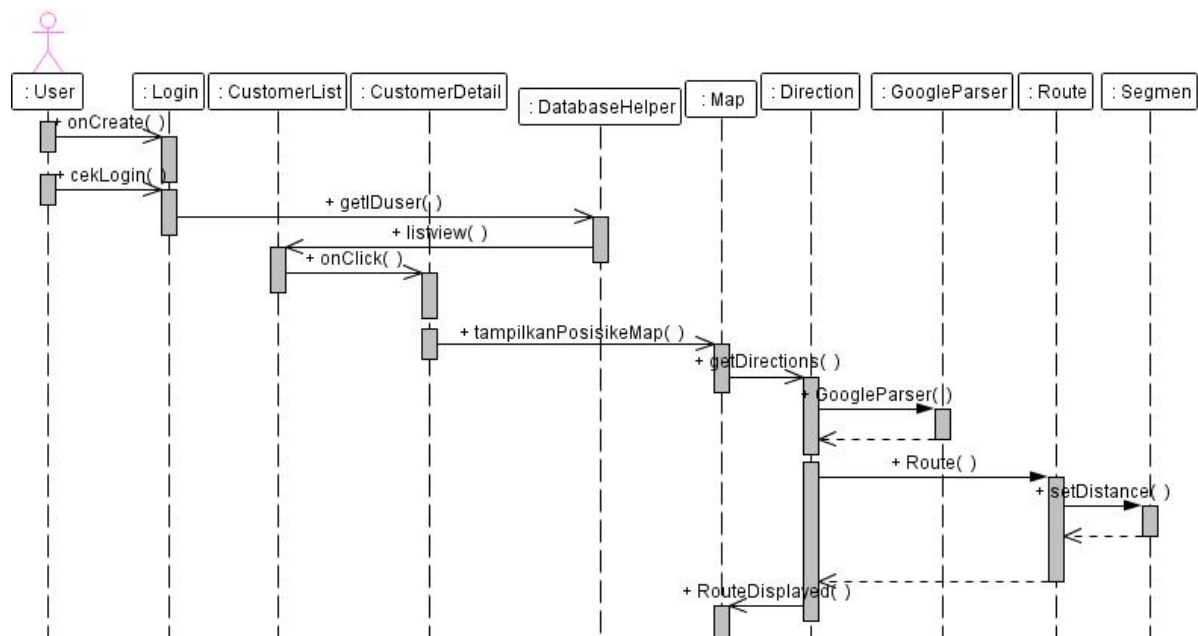
*Activity diagram* merupakan alur kerja pada setiap *usecase*, dimana pada tahap ini akan digambarkan sebuah alur kerja pada aplikasi CCAI. *Activity diagram* pada analisa ini mencakup *activity diagram* setiap *usecase*. Gambar 4.17 berikut ini adalah *activity diagramlogin*. Untuk *activitydiagram* lainnya dapat di lihat pada lampiran B.



Gambar 4.17. Activity Diagram Login

#### 4.1.9.1.5. Sequence Diagram

Untuk menggambarkan interaksi antar objek pada aplikasi CCAI maka dibuatlah *sequence diagram*. *Sequence diagram login* dapat dilihat pada gambar 4.18. Sedangkan *sequence diagram* lainnya dapat dilihat pada Lampiran B.





Gambar 4.18. *Sequence Diagram* Menampilkan Rute ke Lokasi *Customer*

## 4.2. Fase *Elaboration*

### 4.2.1. Perancangan Tabel *Database*

Pada tahap ini dilakukan perancangan Tabel *Database*. Database merupakan bagian penting dalam aplikasi CCAI. Aplikasi CCAI dapat digunakan setelah terkoneksi dengan *database server*, karena semua fitur yang ada pada aplikasi dapat berfungsi apabila selalu terhubung dengan *database server*. *Database server* adalah *database* utama yang ada di server.

#### 4.2.1.1. Perancangan Tabel *User* (Pengguna)

Berikut ini akan dijelaskan rancangan tabel *user* (pengguna):

- Nama : *User*
- Deskripsi : Berisi data-data *user* (pengguna) yang bertugas sebagai pengakses aplikasi
- Primary Key : *idUser*

Tabel 4.5. Tabel *User* (Pengguna)

Nama Field	Type dan Length	Deskripsi	Boleh Null	Default
<i>idUser</i>	int(11)	Identifer	no	<i>PK</i>
UserName	varchar(250)	<i>User Name</i>	no	-

Password	varchar(250)	Kata Sandi	no	-
Nama	varchar(250)	Nama Pengguna	no	<i>FK</i>

#### 4.2.1.2. Perancangan Tabel *Customer* (Pelanggan)

Berikut ini akan dijelaskan rancangan tabel *customer* (pelanggan):

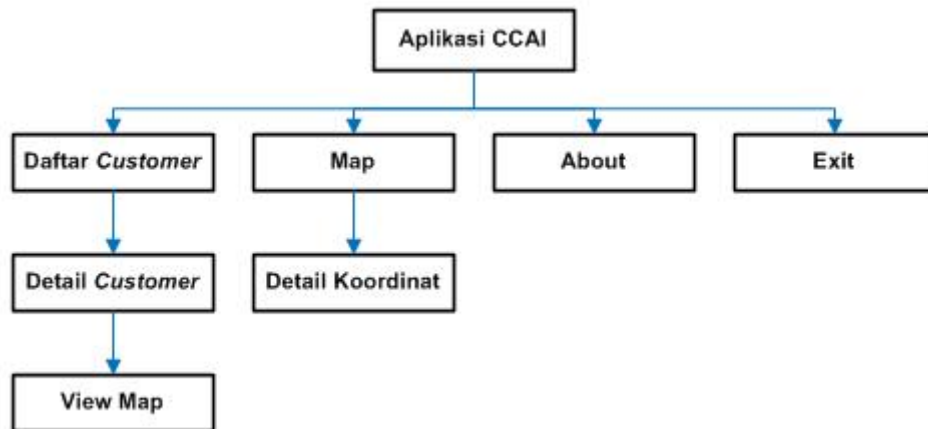
- Nama : *Customer*
- Deskripsi : Berisi data-data *customer* (pelanggan)
- Primary Key : *id*

Tabel 4.6. Tabel *Customer* (Pelanggan)

Nama Field	Type dan Length	Deskripsi	Boleh Null	Default
<i>id</i>	int(11)	Identifer	no	<i>PK</i>
<i>idUser</i>	int(11)	Identifer	no	-
Nama	varchar(250)	Nama <i>Customer</i>	no	<i>FK</i>
Alamat	varchar(250)	Alamat <i>Customer</i>	no	-
<i>Latitude</i>	double	<i>Latitude Customer</i>	no	-
<i>Longitude</i>	double	<i>Longitude Customer</i>	no	-
Gambar	varchar(250)	Foto <i>Customer</i>	no	-

#### 4.2.2. Perancangan Struktur Menu Aplikasi

Rancangan struktur menu merupakan tahapan yang digunakan untuk merancang struktur menu aplikasi CCAI yang akan dibangun. Berikut struktur menu aplikasi dapat dilihat pada gambar 4.19 di bawah ini.



Gambar 4.19. Rancangan Struktur Menu Aplikasi

Seluruh menu pada aplikasi CCAI dapat diakses dengan menekan tombol menu dan *back* yang telah disediakan pada perangkat android untuk menampilkan pilihan menu aplikasi, juga dikombinasikan dengan menyentuh layar untuk melakukan pilihan atau aksi.

#### 4.2.3. Perancangan *Interface* Aplikasi

Rancangan *interface* aplikasi merupakan rancangan antarmuka antara pengguna dengan aplikasi CCAI. Rancangan *interface* ini dibuat agar mempermudah dalam pembuatan antar muka aplikasi pada perangkat android nantinya. Aplikasi CCAI terdiri dari beberapa halaman, antara lain halaman beranda dimana pada halaman ini berisi daftar *customer*, halaman detail *customer*, halaman map, halaman *view maps*, dan halaman *about*. Berikut halaman beranda dapat dilihat pada gambar 4.20 di bawah ini.

##### 4.2.3.1. Perancangan *Interface* Beranda (Daftar *Customer*)

Gambar 4.20 di bawah ini menjelaskan perancangan *interface* beranda pada perangkat Android. Untuk lebih lengkapnya dapat dilihat pada lampiran D.

CCAI	
Daftar Customer:	
1	.....
2	.....
3	.....
4	.....
5	.....
6	.....
7	.....

Gambar 4.20. Perancangan *Interface* Beranda (Daftar *Customer*

## BAB V

### IMPLEMENTASI DAN PENGUJIAN

Pada tahap ini akan dilakukan langkah selanjutnya yang merupakan bagian dari fase *construction* dan *transition*, dimana pada fase *construction* ini penulis akan masuk ketahap pengkodean aplikasi dan implementasi aplikasi. Selanjutnya fase *transition*, pada fase ini dilakukan pengujian aplikasi terhadap *user* menggunakan metode *Blackbox*, kemudian penulis akan mengamati hasil dari pengujian tersebut untuk proses pengambilan kesimpulan dari hasil pengujian.

## **5.1. Fase Construction**

### **5.1.1. Implementasi Aplikasi**

Tahap implementasi ini diperlukan untuk menganalisis kebutuhan-kebutuhan yang diperlukan untuk membangun aplikasi, agar aplikasi yang dibangun sesuai dengan harapan dan fitur-fitur yang dibuat bisa berfungsi sebagaimana mestinya. Pada tahap ini yang akan dibahas meliputi, lingkungan pengembangan, lingkungan implementasi, dan implementasi *interface* aplikasi.

Berikut ini penjelasan tahapan implementasi yang telah dilakukan terhadap aplikasi CCAI.

#### **5.1.1.1. Lingkungan Pengembangan**

Adapun komponen yang penulis gunakan sebagai pendukung proses pengembangan aplikasi meliputi perangkat keras dan perangkat lunak, berikut penjelasannya:

##### **1. Perangkat Keras**

- a. *Processor* : AMD Athlon (tm) 64 X2 Dual Core Processor 50000+
- b. *Memory (RAM)* : 2048 MB

- c. *Harddisk* : 250 GB
  - d. *VGA* : GeForce 6100 PM-M2
2. Perangkat Lunak
- a. *Operating System* (OS) : Windows 7 Ultimate 32-bit
  - b. *Bahasa Pemrograman* : Java dan PHP
  - c. *Tools Pendukung* : IDE Eclipse Indigo, *Java Development Kit* 6u23 (JDK 6u23), *Android Software Development Kit* (SDK), *Android Development Tools* (ADT) Plugins 16.0, *Android Virtual Devices* 2.3.3
  - d. *Browser* : Mozilla Firefox 17.0.1
  - e. *Database Server* : XAMPP (Apache 2, MySQL, PHPMyAdmin)
  - f. *Pemodelan UML* : IBM Rational Rose, Microsoft Visio 2007

#### **5.1.1.2.Lingkungan Implementasi**

Adapun lingkungan implementasi dari aplikasi ini penulis menggunakan perangkat keras dan perangkat lunak yang terdiri dari:

1. Perangkat keras : *Smartphone* Android Samsung Galaxy ACE
2. Perangkat lunak : *Operating System* Android 2.3.5 (*Gingerbread*)

Untuk pendukung proses implementasi ini penulis menggunakan jasa penyedia layanan *hosting* dengan *domain* “http://gubukkayangan.com”.

#### **5.1.1.3.Implementasi *Interface* Aplikasi**

Pada tahap ini akan ditampilkan beberapa *interface* dari aplikasi CCAI. Adapun hasil dari implementasi untuk halaman *login* pada perangkat Android

dapat dilihat pada gambar 5.1 dibawah ini. Selanjutnya untuk hasil implementasi lainnya dapat dilihat pada lampiran D.



Gambar 5.1. Hasil Implementasi Halaman *Login*

## 5.2. Fase *Transition*

### 5.2.1. Pengujian Aplikasi

Pengujian aplikasi pada tugas akhir ini dilakukan untuk menemukan kesalahan atau kekurangan yang terdapat pada aplikasi yang telah dibangun. Pengujian menggunakan dua metode, yaitu *black box* dan *user acceptance test*. Adapun pengujian dengan metode *black box* yaitu pengujian terhadap cara kerja dan fitur aplikasi, sedangkan pengujian *user acceptance test* dilakukan untuk mengetahui tingkat kepuasan *user* dengan aplikasi.

#### 5.2.1.1. Pengujian *Black Box*

Pengujian *black box* ini dilakukan untuk mengetahui apakah aplikasi CCAI yang telah dibangun sesuai dengan kebutuhan. Pengujian yang dilakukan meliputi tiga jenis pengujian, diantaranya:

1. Pengujian fitur dan fungsi aplikasi
2. Pengujian aplikasi dengan beberapa *provider* jaringan dan pada beberapa merk perangkat Android
3. Pengujian aplikasi untuk menampilkan rute jalan ke lokasi *customer*

#### **5.2.1.1.1. Pengujian Fitur dan Fungsi Aplikasi**

Pengujian menggunakan perangkat android Samsung Galaxy ACE. Adapun hasil dari pengujian *black box* untuk fitur dan fungsi aplikasi CCAI dapat dilihat pada tabel 5.1.



Table 5.1. Pengujian Fitur dan Fungsi Aplikasi CCAI

No.	Nama Pengujian	Deskripsi	Prekondisi	Prosedur Pengujian	Aksi/ Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
1	Akses Aplikasi dan masuk ke halaman <i>login</i>	Pengujian untuk mengakses aplikasi dan masuk ke halaman login	-	Klik logo aplikasi pada perangkat android	Klik logo	Menampilkan halaman <i>login user</i>	Tampil halaman <i>login user</i>	Benar
2	Status <i>login</i> dan menampilkan daftar <i>customer</i>	Pengujian untuk status <i>login</i> dan aplikasi menampilkan daftar <i>customer</i>	Tampil halaman <i>login user</i>	Klik tombol <i>login</i> pada aplikasi	Klik <i>login</i>	Menampilkan daftar <i>customer</i>	Tampil daftar <i>customer</i>	Benar
3	Menampilkan detail <i>customer</i>	Pengujian untuk menampilkan detail <i>customer</i>	Tampil halaman daftar <i>customer</i>	Klik salah satu nama <i>customer</i> dari daftar	Klik nama <i>customer</i>	Menampilkan detail <i>customer</i>	Tampil detail <i>customer</i>	Benar

			(Beranda)	<i>customer</i>				
4	Menampilkan Map Rute Jalan	Pengujian untuk menampilkan map rute jalan	Tampil halaman detail <i>customer</i>	Klik tombol <i>view</i> map yang terdapat pada halaman detail <i>customer</i>	Klik <i>view</i> map	Menampilkan map rute jalan	Tampil map rute jalan	Benar
5	Menampilkan Map	Pengujian untuk menampilkan map	Tampil halaman daftar <i>customer</i> (Beranda)	Klik tombol menu yang terdapat pada perangkat android, lalu klik map pada opsi	Klik menu, klik map	Menampilkan Map	Tampil Map	Benar

				menu				
6	Menampilkan <i>About</i>	Pengujian untuk menampilkan <i>about</i>	Tampil halaman daftar <i>customer</i>  (Beranda)	Klik tombol menu yang terdapat pada perangkat android, lalu klik <i>about</i> pada opsi menu	Klik menu, klik <i>about</i>	Menampilkan <i>About</i>	Tampil <i>About</i>	Benar

#### 5.2.1.1.2. Pengujian Aplikasi dengan beberapa *Provider* Jaringan dan Pada beberapa Merk Perangkat Android

Adapun hasil dari pengujian *blackbox* untuk aplikasi dengan beberapa *provider* jaringan dan pada beberapa merk perangkat Android dapat dilihat pada tabel 5.2.

Tabel 5.2. Pengujian Aplikasi dengan beberapa *Provider* Jaringan dan Pada beberapa Merk Perangkat Android

No.	Tanggal - Pukul	Perangkat Android	Versi Android	<i>Provider</i>	Pengujian	Hasil
1	22 Desember 2012 – 16.13	Samsung Galaxy Ace	Gingerbread (2.3)	3 (Tri)	Koneksi Ke <i>Database Server</i>	Berhasil
					<i>Login</i>	Berhasil
					Daftar <i>Customer</i>	Berhasil
					Detail <i>Customer</i>	Berhasil
					Map	Berhasil
					<i>About</i>	Berhasil
2	23 Desember 2012 – 14.21	Samsung Galaxy Fit	Froyo (2.2)	As (T-sel)	Koneksi Ke <i>Database Server</i>	Berhasil
					<i>Login</i>	Berhasil
					Daftar <i>Customer</i>	Berhasil
					Detail <i>Customer</i>	Berhasil
					Map	Berhasil
					<i>About</i>	Berhasil
3	24 Desember 2012 – 09.12	Samsung Galaxy Y	Gingerbread (2.3)	XL	Koneksi Ke <i>Database Server</i>	Berhasil
					<i>Login</i>	Berhasil
					Daftar <i>Customer</i>	Berhasil


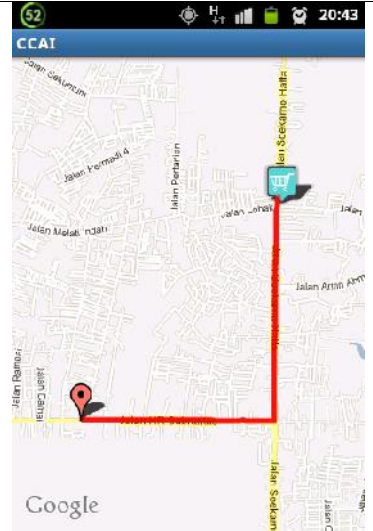
					Detail <i>Customer</i>	Berhasil
					Map	Berhasil
					About	Berhasil
4	24 Desember 2012 – 11.46	Samsung Galaxy Gio	Froyo (2.2)	As (T-sel)	Koneksi Ke <i>Database Server</i>	Berhasil
					Login	Berhasil
					Daftar <i>Customer</i>	Berhasil
					Detail <i>Customer</i>	Berhasil
					Map	Berhasil
					About	Berhasil
5	24 Desember 2012 – 19.29	Sony Ericson Xperia Mini Pro	Froyo (2.2)	Simpati (T-sel)	Koneksi Ke <i>Database Server</i>	Berhasil
					Login	Berhasil
					Daftar <i>Customer</i>	Berhasil
					Detail <i>Customer</i>	Berhasil
					Map	Berhasil
					About	Berhasil

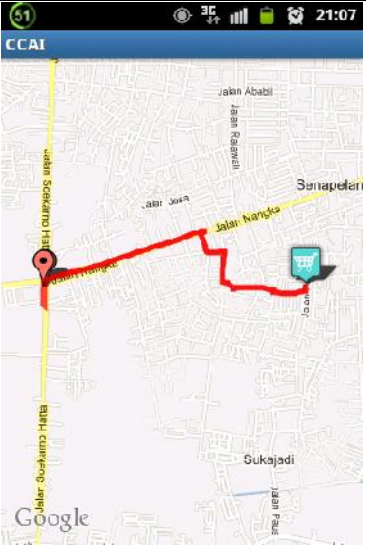
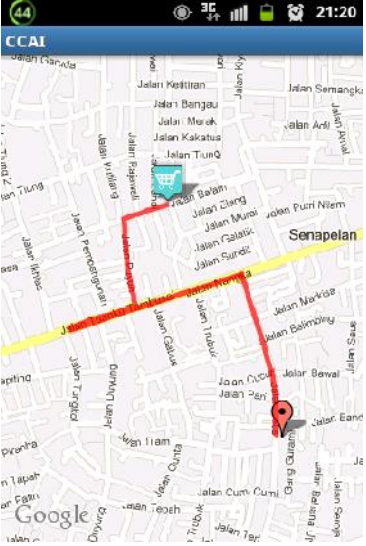
#### 5.2.1.1.3. Pengujian Aplikasi untuk Menampilkan Rute Jalan ke Lokasi Tujuan

Berikut akan ditampilkan hasil dari pengujian aplikasi untuk menampilkan rute jalan ke lokasi tujuan yang dapat dilihat pada tabel 5.3.

Tabel 5.3. Pengujian Aplikasi untuk Menampilkan Rute Jalan ke Lokasi *Customer*



No.	Posisi Awal <i>User</i>	Lokasi Tujuan	Hasil Map


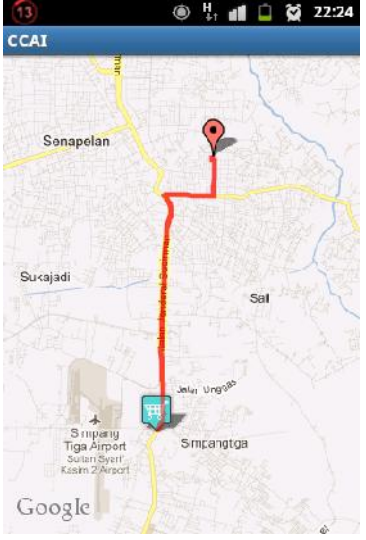
1	PT. CCAI	Central Panam Elektronik	
2	Central Panam Elektronik	Amazone SKA	

3	Amazone SKA	AA Mart	
4	AA Mart	Al Barokah Mart	





7	Elang Futsal	Hotel Dharma Utama	 <p>A Google Map screenshot showing a route from a red pin on Jalan Jendral Ahmad Yani to a blue pin on Jalan Pangeran Hidayat. The route is marked with a red line. The map includes labels for various streets and landmarks, such as Jalan Agus Salim, Jalan Pangeran Hidayat, and Jalan Jendral Sudirman.</p>
8	Hotel Dharma Utama	Internasional Club MP	 <p>A Google Map screenshot showing a route from a red pin on Jalan Jendral Sudirman to a blue pin on Jalan Heng Tiah. The route is marked with a red line. The map includes labels for various streets and landmarks, such as Jalan Teuku Umar, Jalan Heng Tiah, and Jalan Jendral Sudirman.</p>

9	Internasional Club MP	Mini Market Adelia	
10	Mini Market Adelia	Aziziah Swalayan	

#### 5.2.1.2. User Acceptance Test

Pengujian *User Acceptance Test* merupakan pengujian yang mana berisi pertanyaan-pertanyaan mengenai aplikasi yang telah dibangun serta fitur yang terdapat pada aplikasi. Adapun hasil dari pada pengujian tersebut akan dilampirkan berupa kuisioner yang diisi oleh *user*. Jumlah responden *user* adalah 10 orang dengan jumlah pertanyaan 5.

Adapun kategori penilaian yang diberikan untuk kuisioner adalah sebagai berikut:

1. Sangat Tidak Bagus (STB)

2. Tidak Bagus (TB)
3. Biasa Saja (BS)
4. Bagus (B)
5. Sangat Bagus (SB)

Pertanyaan pada kuisisioner yang akan diajukan pada *user* adalah sebagai berikut:

Tabel 5.4. Kuisisioner *User*

No.	Pertanyaan	STB	TB	BS	B	SB
1	Desain <i>interface</i> aplikasi CCAI					
2	Aplikasi dapat memberikan informasi daftar <i>customer</i> yang akan dituju					
3	Aplikasi dapat memberikan informasi lokasi <i>customer</i>					
4	Aplikasi dapat memberikan informasi rute jalan menuju lokasi <i>customer</i>					
5	Kemudahan menggunakan Aplikasi					

Setelah mendapatkan hasil dari kuisisioner yang telah diisi oleh *user*, selanjutnya data dari kuisisioner tersebut diolah untuk mendapatkan hasil dari penilaian *user acceptance test*. Adapun hasil penilaian *user acceptance test* tersebut adalah sebagai berikut:

Tabel 5.5. Hasil Pengujian *User Acceptance Test*

Pertanyaan	Jawaban <i>User</i>									
	STB	%	TB	%	BS	%	B	%	SB	%
1	0	0 %	2	20 %	2	20 %	4	40 %	2	20 %

2	0	0 %	0	0 %	1	10 %	7	70 %	2	20 %
3	0	0 %	0	0 %	5	50 %	4	40 %	1	10 %
4	0	0 %	0	0 %	0	0 %	4	40 %	6	60 %
5	0	0 %	0	0 %	3	30 %	6	60 %	1	10 %
<b>Total</b>	<b>0</b>	<b>0 %</b>	<b>2</b>	<b>4 %</b>	<b>11</b>	<b>22 %</b>	<b>25</b>	<b>50 %</b>	<b>12</b>	<b>24 %</b>

Cara perhitungan:

- *User* 10
- Pertanyaan 5

Setiap satu pertanyaan memiliki 5 jawaban yang harus dipilih salah satunya oleh *user*. Jawaban yang dipilih oleh satu *user* akan bernilai 1 dengan persentase 10 %.

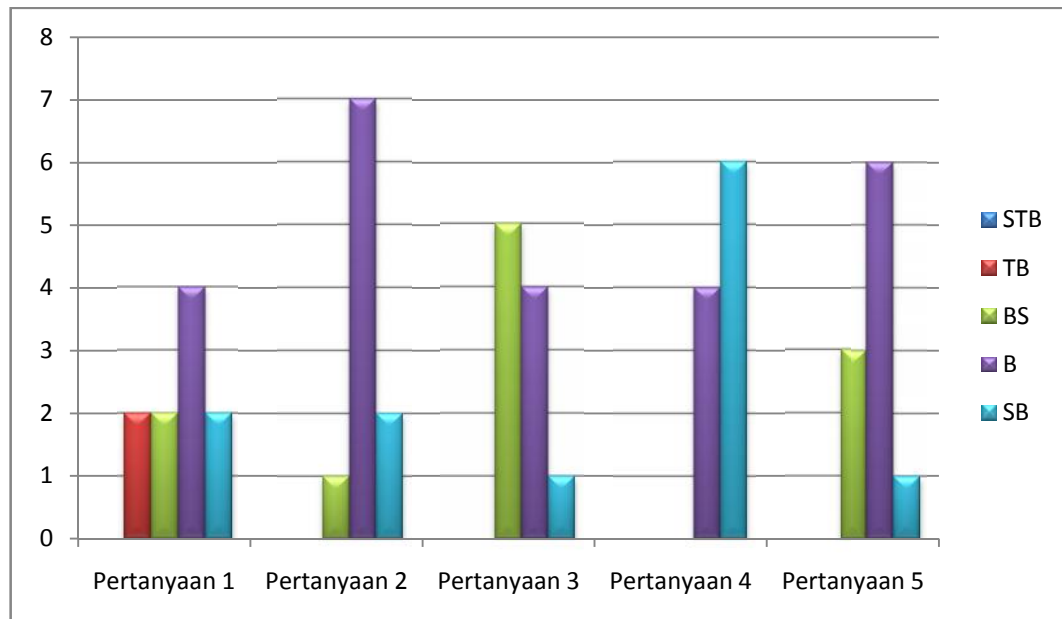
Contoh :

Lihat tabel 5.5 pada kolom jawaban TB (Tidak Bagus), untuk pertanyaan 1 ada 2 *user* yang memilih jawaban TB sehingga bernilai 2 dengan persentase 20 %.

Rumus untuk mendapatkan total persentase sebagai berikut:

Total Persentase = Jumlah Persentase : Total Pertanyaan

Untuk hasil pengujian *user acceptance test* menggunakan kuisioner dapat dilihat pada lampiran E. Selanjutnya dari tabel 5.5 dapat digambarkan dalam bentuk grafik seperti dibawah ini:



Gambar 5.2. Grafik Pengujian *User Acceptance Test*

Kemudian dari grafik pengujian *user acceptance test* dapat diambil kesimpulan sebagai berikut:

1. *User* yang melakukan pengujian aplikasi memilih Sangat Tidak Bagus (STB) dengan persentase 0 %
2. *User* yang melakukan pengujian aplikasi memilih Tidak Bagus (TB) dengan persentase 4 %
3. *User* yang melakukan pengujian aplikasi memilih Biasa Saja (BS) dengan persentase 22 %
4. *User* yang melakukan pengujian aplikasi memilih Bagus (B) dengan persentase 50 %
5. *User* yang melakukan pengujian aplikasi memilih Sangat Bagus (SB) dengan persentase 24 %

### 5.2.2. Kesimpulan Pengujian

Adapun kesimpulan yang dapat diambil dari hasil pengujian *black box* dan *user acceptance test* aplikasi adalah sebagai berikut:

1. Aplikasi CCAI yang telah dibangun sesuai dengan yang dianalisis dan dirancang sebelumnya.
2. Koneksi antara aplikasi CCAI dan *database server* berhasil.

3. Pada perangkat android dan *provider* yang berbeda yang telah dilakukan pengujian, aplikasi CCAI berhasil digunakan.
4. Aplikasi CCAI berhasil menampilkan map rute jalan dari lokasi awal *user* ke lokasi tujuan.
5. Secara keseluruhan berdasarkan hasil pengujian *user acceptance test* di atas dapat penulis ambil kesimpulan bahwa aplikasi CCAI ini dapat diterima dengan baik oleh *user* yang dalam hal ini sebagai supir PT. CCAI.

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Kesimpulan yang dapat penulis ambil dari hasil penelitian tugas akhir ini adalah sebagai berikut:

1. Aplikasi CCAI yang dibangun sesuai dengan studi kasus yang penulis lakukan di PT. Coca Cola Amatil Indonesia Sales Office Pekanbaru, dalam hal layanan pengantaran produk ke lokasi *customer*.
2. Aplikasi yang dibangun sesuai dengan analisa dan perancangan sebelumnya, aplikasi berhasil koneksi ke *server*, dan berhasil menampilkan fitur dan konten yang terdapat pada server.
3. Berdasarkan pada tabel 5.3 aplikasi berhasil menampilkan rute jalan ke lokasi *customer*.
4. Aplikasi dapat memberikan informasi jarak tempuh dan waktu tempuh.
5. Aplikasi menampilkan urutan daftar *customer* berdasarkan perhitungan algoritma Dijkstra dan kriteria yang digunakan untuk menentukan rute terpendek adalah jarak garis lurus antar lokasi, selanjutnya Google Maps akan menampilkan rute jalan untuk menuju ke lokasi masing-masing *customer*, sehingga pada situasi tertentu aplikasi tidak dapat memberikan solusi rute dan jarak yang optimal.
6. Google Maps yang digunakan belum bisa memfasilitasi fitur untuk memperoleh informasi jalan satu arah dan jalan yang tidak bisa dilewati.

## 6.2 Saran

Berikut ini beberapa saran yang penulis sampaikan untuk pengembangan aplikasi CCAI selanjutnya, adalah sebagai berikut:

1. Untuk pengembangan tugas akhir selanjutnya dapat membahas dari sisi server, sehingga admin bisa memantau kinerja *user* melalui sistem.
2. Aplikasi CCAI untuk selanjutnya dapat digunakan pada multi *platform*.
3. Untuk pengembangan selanjutnya waktu tempuh diperoleh dengan mempertimbangan kondisi jalan dan kemacetan.
4. Aplikasi CCAI menggunakan algoritma yang efisien untuk permasalahan *Traveling Salesman Problem* (TSP) untuk memperoleh hasil yang optimal.



## DAFTAR PUSTAKA

- Afwani, R., dan Fiarni, C., “*Perancangan location based service untuk mobile cloud reminder system Penyakit Tuberkulosis di Indonesia*”, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2011
- Azfar, A., Hossen, Md.S., Marval, M.J.P., dan Khan, R.H., “*Location-based Services Using SIP*”, International Journal of Engineering Science and Technology, 2010
- Beauburn, R., Moulin, B., dan Jabeur, N., “*An architecture for delivering location-based services*”, International Journal of Computer Science and Network Security, 2007
- Booch, G., “*Object-oriented Analysis and Design*”, Santa Clara, California, 1998
- Cho, H.J., dan Chung, C.W., “*An efficient and scalable approach to CNN Queries in a Road Network*”, VLDB, 2005
- Dewi, L.J.E., “*Pencarian Rute Terpendek Tempat Wisata di Bali dengan menggunakan Algoritma Dijkstra*”, SNATI, 2010
- Dharwiyanti, S., dan Wahono, R.S., “*Pengantar Unified Modeling Language (UML)*”, Kuliah Umum IlmuKomputer.Com, 2003
- Faizah, I., “*Rancang Bangun Perangkat Lunak Penentuan Rute Perjalanan Wisata di Malang menggunakan Algoritma Dijkstra*”, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 2010
- Nazaruddin, S.H., “*Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*”, Informatika, Bandung, 2011
- IAMAI, “*Location Based Services (LBS) on Mobile in India*”, 2008.[www.indicus.net](http://www.indicus.net)
- Jenkins, M., “*Introduction to Route Calculation*”, NAVTEQ Principal Consultant, 2007
- Kruchten, P., “*What is The Rational Unified Process?*”, Rational Fellow, Rational Software Canada, 2001
- Kupper, A., “*Location-based Service: Fundamentals and Operation*”, Ludwig Maxmilian University, Munich, Germany, 2005

- Kushwaha, A., dan Kushwaha, V., "*Location Based Services using Android Mobile Operating System*", International Journal of Advances in Engineering and Technologi, 2011
- Mabrouk, M.ESRI., "*OpenGIS Location Services (OpenLS): Core Services*", Open Geospatial Internasional, 2003
- Munir, R., "*Matematika Diskrit*", Informatika, Bandung, 2005
- Rational Team, "*Rational Unified Process*", BPSDT, 2001
- Rutter, S.J., "*Dijkstra's Algorithm Final Project*", EDUC, 2009
- Shiode, N., Li, C., Batty, M., Longley, P., dan Maguire, D., "*The Impact and Penetration of Location-based Services*", CASA, 2002
- Sniedovich, M., "*Dijkstra's algorithm revisited: the dynamic programming connexion*", University of Melbourne, Australia, 2006
- Suprianto, D., dan Agustina, R., "*Pemrograman Aplikasi Android*", Media Kom, Yogyakarta, 2012
- Wahyuningrum, E., Sari, R.W., Wiguna, S.P., Umuhani, U., Wicaksana, I.W.S., "*Software Engineering Methodology Rational Unified Process*", 2011
- Wang, B., "*Mobile Location-based Services in New Zealand*", Faculty of Design and Creative Technologies, Auckland University of Technology, 2008
- Yulia, K.G., Tanuhardja, J., "*Perencanaan Rute Perjalanan di Jawa Timur dengan Dukungan GIS Menggunakan Metode Dijkstra's*", Jurnal Informatika, 2002
- \_\_\_\_\_, "*Dijkstra Algorithm*", [available] online [http://www.enotes.com/topic/Dijkstra%27s\\_algorithm](http://www.enotes.com/topic/Dijkstra%27s_algorithm), diakses tanggal 21 Mei 2012
- \_\_\_\_\_, "*Google Maps APP*", [available] online <http://www.code.google.com/apis/maps/documentation/javascript>, 2012, diakses tanggal 13 Desember 2012
- \_\_\_\_\_, "*Location Based Services: Considerations and Challenges*", [www.northstream.se](http://www.northstream.se), 2001



## **LAMPIRAN A**

## Deskripsi Rinci *Use Case Specification*

### A.1 *Usecase Specification* Menampilkan Map

Tabel A.1. *Usecase Specification* Menampilkan Map

<i>Use Case Name</i>	Menampilkan Peta Pelanggan
<i>Deskripsi Singkat</i>	<i>User</i> dapat menampilkan halaman map dengan menggunakan aplikasi untuk melihat <i>customer</i> yang akan dikunjungi
<i>Aktor</i>	<i>User</i>
<i>Pre Condition</i>	<i>User</i> harus melakukan login untuk mengakses aplikasi
<i>Post Condition</i>	Aplikasi menampilkan map beserta titik koordinat <i>customer</i> yang telah ditandai
<i>Normal Flow</i>	<i>User</i> melakukan <i>login</i> untuk mengakses aplikasi kemudian <i>user</i> dapat menampilkan halaman map dengan menggunakan aplikasi
<i>Alternate Flow</i>	Seandainya <i>user</i> ingin melihat lebih detail informasi titik koordinat masing-masing <i>customer</i> maka diklik titik koordinat <i>customer</i> yang telah ditandai

### A.2 *Usecase Specification* Login

Tabel A.2. *Usecase Specification Login*

<i>Use Case Name</i>	<i>Login</i>
<i>DeskripsiSingkat</i>	Aplikasi memverifikasi pengguna untuk masuk dan mengakses aplikasi
<i>Aktor</i>	<i>User</i>
<i>Pre Condition</i>	Memiliki nama dan password
<i>Post Condition</i>	<i>User</i> berhasil mendapatkan hak akses pada aplikasi
<i>Normal Flow</i>	<i>User</i> menginputkan nama dan passwordnya untuk dapat mengakses aplikasi
<i>Alternate Flow</i>	-

### **A.3 Usecase Specification Menampilkan Detail Customer**

Tabel A.3. *Usecase Specification Menampilkan Detail Customer*

<i>Use Case Name</i>	Menampilkan Informasi Detail Pelanggan
<i>DeskripsiSingkat</i>	<i>User</i> dapat menampilkan halaman detail <i>customer</i> untuk memperoleh informasi lengkap tentang <i>customer</i>
<i>Aktor</i>	<i>User</i>
<i>Pre Condition</i>	<i>User</i> melihat daftar <i>customer</i>
<i>Post Condition</i>	Aplikasi menampilkan informasi detail <i>customer</i> yang berisi tentang foto, nama, alamat, dan nomor telepon <i>customer</i>

<i>Normal Flow</i>	<i>User</i> melakukan <i>login</i> untuk mengakses aplikasi kemudian aplikasi menampilkan halaman daftar <i>customer</i> , lalu <i>user</i> mengklik salah satu daftar <i>customer</i> kemudian aplikasi akan menampilkan detail <i>customer</i> terpilih berupa foto, nama, alamat, dan nomor telepon <i>customer</i>
<i>Alternate Flow</i>	Seandainya <i>user</i> ingin melihat rute jalan menuju lokasi <i>customer</i> terpilih, maka <i>user</i> harus menekan tombol <i>view map</i> yang terdapat pada halaman detail <i>customer</i>

#### **A.4 Usecase Specification Menampilkan Informasi Koordinat**

Tabel A.4. Usecase Specification Menampilkan Informasi Koordinat

<i>Use Case Name</i>	Menampilkan Informasi Koordinat Pelanggan
<i>Deskripsi Singkat</i>	<i>User</i> dapat melihat informasi koordinat <i>customer</i> yang akan ditampilkan aplikasi
<i>Aktor</i>	<i>User</i>
<i>Pre Condition</i>	<i>User</i> menampilkan map
<i>Post Condition</i>	Aplikasi Menampilkan Informasi koordinat <i>customer</i> yang berisitentang nama dan alamat <i>customer</i>
<i>Normal Flow</i>	<i>User</i> melakukan <i>login</i> untuk mengakses aplikasi kemudian <i>user</i> dapat menampilkan halaman map, lalu <i>user</i> mengklik salah satu titik koordinat <i>customer</i> yang telah ditandai pada map kemudian aplikasi akan

	menampilkan informasi koordinat <i>customer</i> berupa nama dan alamat <i>customer</i>
<i>Alternate Flow</i>	-

### A.5 Usecase Specification Menampilkan Map Rute Jalan

Tabel A.5. *Usecase Specification* Menampilkan Map Rute Jalan

<i>Use Case Name</i>	Menampilkan Map Rute Jalan
<i>Deskripsi Singkat</i>	<i>User</i> melihat map rute jalan menuju lokasi <i>customer</i> yang ditampilkan aplikasi
<i>Aktor</i>	<i>User</i>
<i>Pre Condition</i>	<i>User</i> melihat halaman detail <i>customer</i>
<i>Post Condition</i>	Aplikasi menampilkan map rute jalan menuju lokasi <i>customer</i>
<i>Normal Flow</i>	<i>User</i> berada pada halaman detail <i>customer</i> kemudian <i>user</i> menekan tombol <i>view map</i> yang terdapat pada halaman detail <i>customer</i> , lalu aplikasi akan menampilkan map berupa rute jalan menuju lokasi <i>customer</i>
<i>Alternate Flow</i>	Seandainya <i>user</i> ingin melihat informasi jarak tempuh dan waktu tempuh, maka <i>user</i> harus menekan koordinat lokasi tujuan





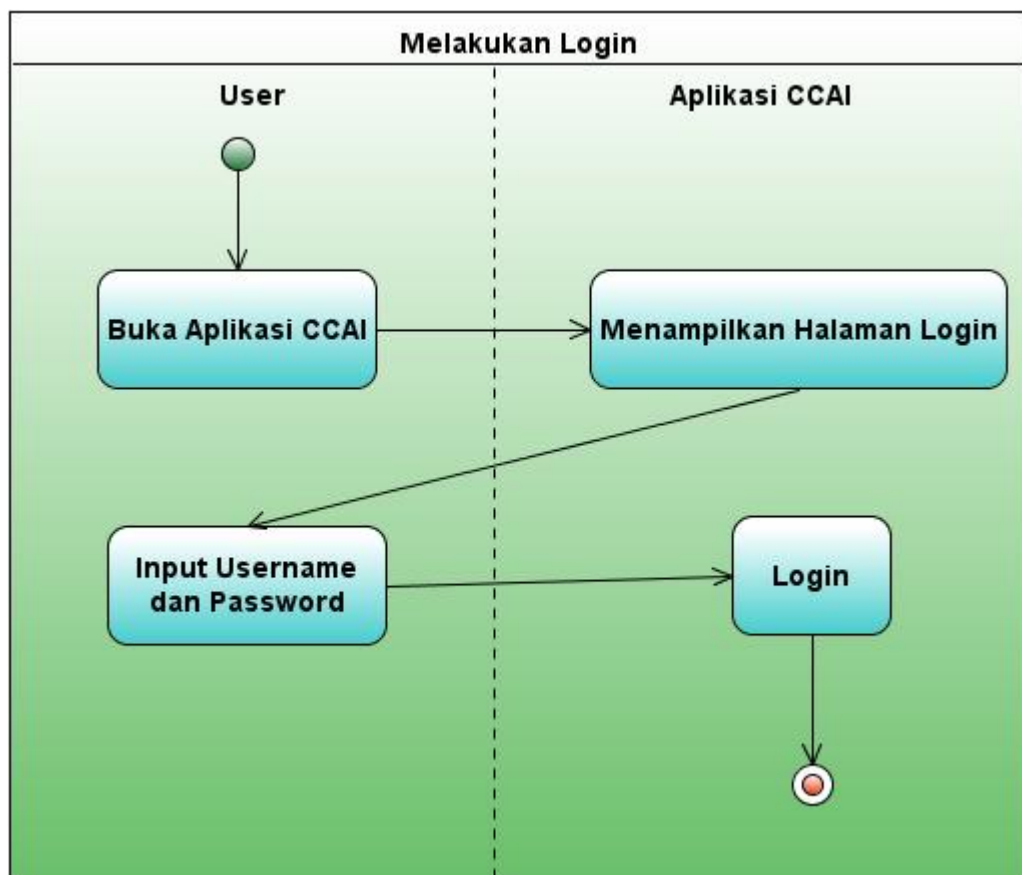
## **LAMPIRAN B**

## Deskripsi Rinci *ACTIVITY DIAGRAM*

### B.1. *Activity Diagram*

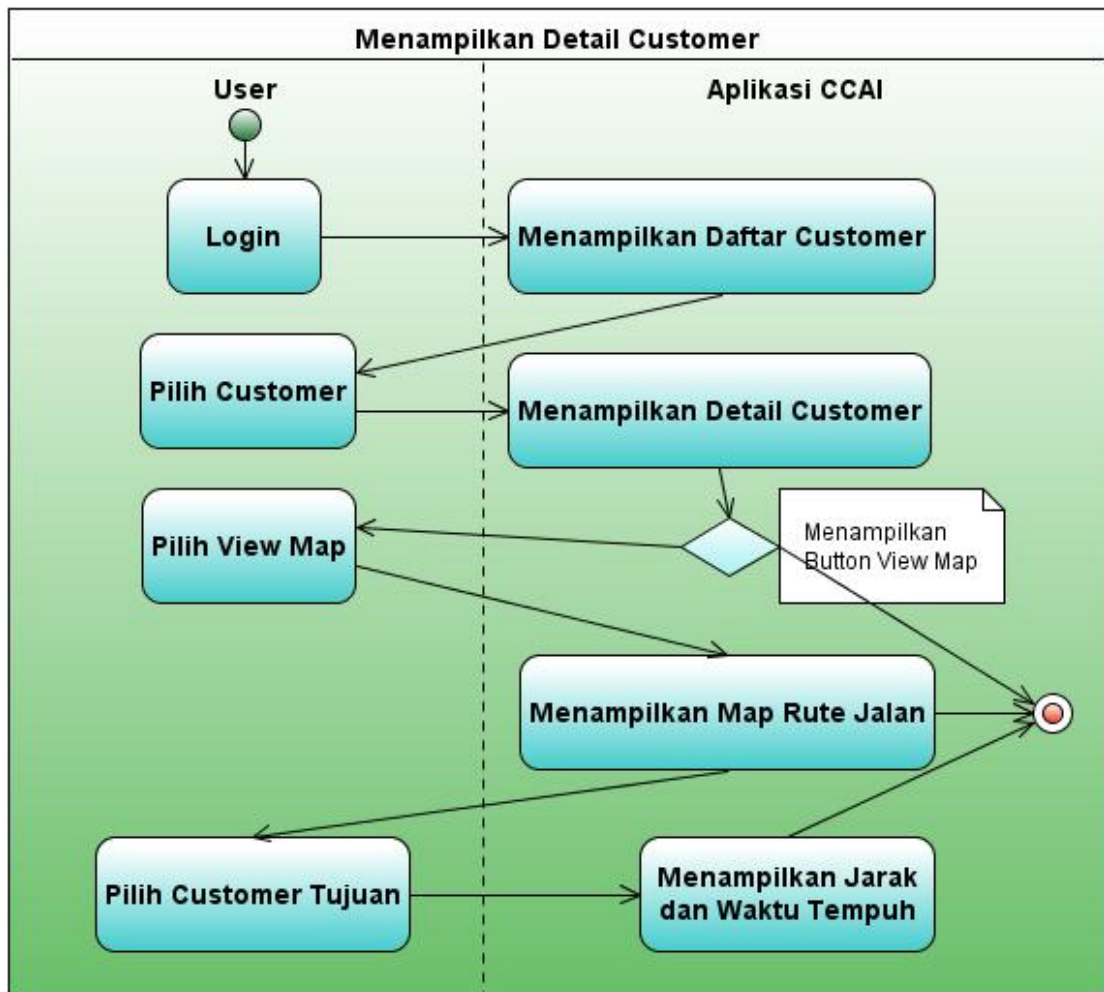
*Activity diagram* merupakan alur kerja pada setiap *usecase*, dimana pada tahap ini akan digambarkan sebuah alur kerja pada aplikasi CCAI, berikut ini beberapa *activity diagram* seperti, *activity diagram login*, *activity diagram detail customer*, *activity diagram map*, *activity diagram about*.

#### B.1.1. *Activity Diagram Login*



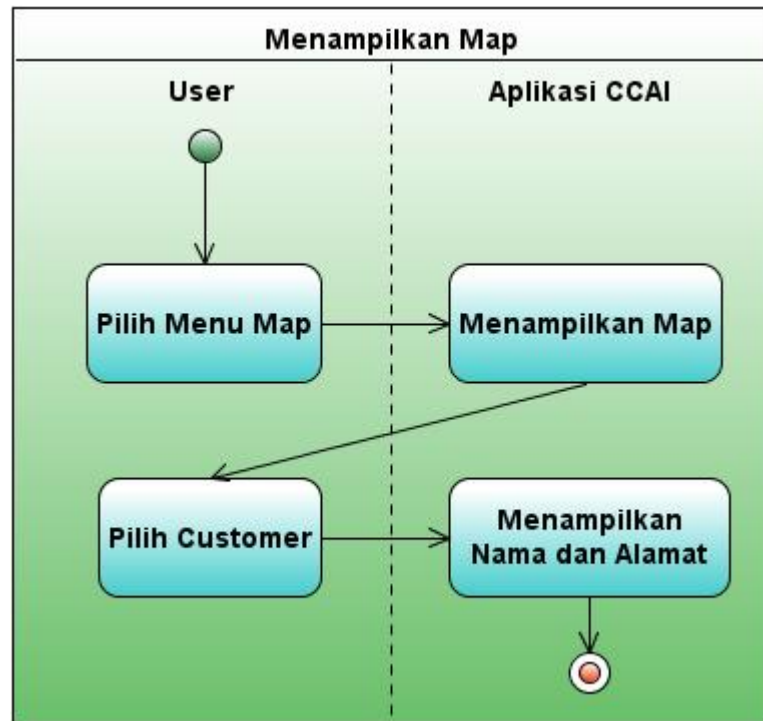
Gambar B.1. *Activity Diagram Login*

#### B.1.2. *Activity Diagram Detail Customer*



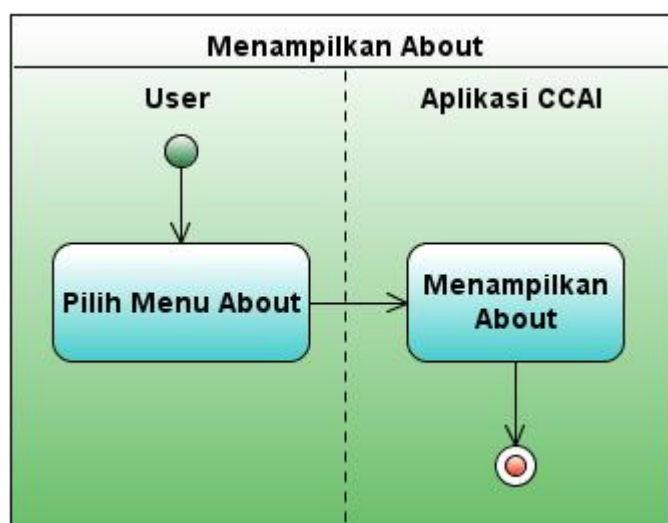
Gambar B.2. Activity Diagram Detail Customer

### B.1.3. Activity Diagram Map



Gambar B.3. Activity Diagram Map

### B.1.4. Activity Diagram About



Gambar B.4. Activity Diagram About

## **LAMPIRAN C**

## Deskripsi Rinci RANCANGAN *INTERFACE* APLIKASI

### C.1. Perancangan *Interface* Aplikasi

Rancangan *interface* aplikasi merupakan rancangan antarmuka antara pengguna dengan aplikasi CCAI. Rancangan *interface* ini dibuat agar mempermudah dalam pembuatan antar muka aplikasi pada perangkat android nantinya. Aplikasi CCAI terdiri dari beberapa halaman, antara lain halaman beranda dimana pada halaman ini berisi daftar *customer*, halaman detail *customer*, halaman map, halaman *view maps*, dan halaman *about*.

#### C.1.1. Perancangan *Interface* Beranda (Daftar *Customer*)

CCAI	
Daftar Customer:	
1	.....
2	.....
3	.....
4	.....
5	.....
6	.....
7	.....

Gambar C.1. Perancangan *Interface* Beranda (Daftar *Customer*)

#### C.1.2. Perancangan *Interface* Detail *Customer*

CCAI

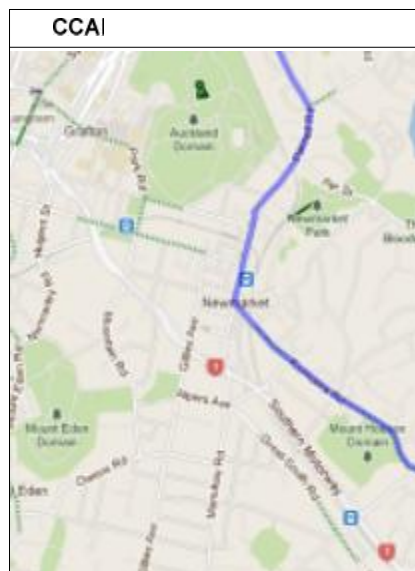
Gambar  
Toko Customer

Nama Customer

Alamat Customer

View  
Map

### C.1.3. Perancangan *InterfaceMap*





#### C.1.4. Perancangan *InterfaceDirection*



Gambar C.4. Perancangan *InterfaceDirection*

#### C.1.5. Perancangan *InterfaceAbout*



Gambar C.5. Perancangan *InterfaceAbout*



## **LAMPIRAN D**

## hasil implementasi

### D.1. Implementasi *Interface* Aplikasi

Berikut akan ditampilkan beberapa implementasi *interface* dari aplikasi CCAI.

#### D.1.1. Implementasi Halaman *Login*



Gambar D.1. Hasil Implementasi Halaman *Login*

### D.1.2. Implementasi Halaman Beranda (Daftar *Customer*)



Gambar D.2. Hasil Implementasi Halaman Beranda (Daftar *Customer*)

### D.1.3. Implementasi Halaman Detail *Customer*



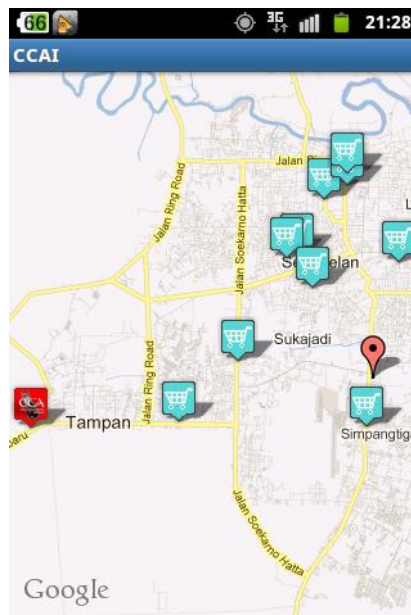
Gambar D.3. Hasil Implementasi Halaman Detail *Customer*

#### D.1.4. Implementasi Halaman *View Maps*



Gambar D.4. Hasil Implementasi Halaman *View Maps*

#### D.1.5. Implementasi Halaman Map



Gambar D.5. Hasil Implementasi Halaman Map

#### D.1.6. Implementasi Halaman *About*



Gambar D.6. Hasil Implementasi Halaman *About*

## **LAMPIRAN E**



## KUISIONER PENGUJIAN USER ACCEPTANCE TEST APLIKASI CCAI

Berikut ini kuisionerpengujian *user acceptance test* aplikasi CCAI.

Nama : .....

NIK : .....

Jabatan : .....

Berilah tanda ceklis ( ) pada jawaban anda terhadap pertanyaan yang akan diajukan berikut ini:

No.	Pertanyaan	STB	TB	BS	B	SB
1	Desain <i>interface</i> aplikasi CCAI					
2	Aplikasi dapat memberikan informasi daftar <i>customer</i> yang akan dituju					
3	Aplikasi dapat memberikan informasi lokasi <i>customer</i>					
4	Aplikasi dapat memberikan informasi rute jalan menuju lokasi <i>customer</i>					
5	Kemudahan menggunakan Aplikasi					

Kritik dan saran jika ada:

----------------------

Terima kasih atas ketersediaan anda dalam mengisi kuisioner ini.

Tanda Tangan,

( )

**DATA PRIBADI:**

Nama : Luthfi Fahronzi  
Jenis Kelamin : Laki – Laki  
Tempat, Tanggal Lahir : Durai, 20  
September 1989  
Agama : Islam  
Status : Belum Menikah  
Alamat : KP. Tengah Durai,  
Kec. Durai  
No. Handphone : 081268222993  
E-mail :  
luthfifahronzi@gm  
ail.com

**PENDIDIKAN FORMAL:**

- 1995 – 2001 : SD Negeri 002 Durai
- 2001 – 2004 : SMP Negeri 2 Moro ( Kelas Jauh)
- 2004 - 2007 : SMA Negeri 3 Kundur
- 2007 – 2013 : Universitas Islam Negeri Sultan Syarif Kasim Pekanbaru Riau

Pekanbaru, 03 April 2013

Hormat saya,

**LUTHFI FAHRONZI**

